INSTITUTE FOR BIOMEDICAL ENGINEERING AND INFORMATICS

DEPARTMENT OF COMPUTER SCIENCE AND AUTOMATION

TECHNISCHE UNIVERSITÄT ILMENAU

Dissertation

submitted by

M.Sc. David Heinemann

submitted on November 15, 2019

urn:nbn:de:gbv:ilm1-2020000175

INSTITUTE FOR BIOMEDICAL ENGINEERING AND INFORMATICS

DEPARTMENT OF COMPUTER SCIENCE AND AUTOMATION

TECHNISCHE UNIVERSITÄT ILMENAU

Dissertation

zur Erlangung des akademischen Grades Doktoringenieur (Dr.-Ing.)

3D based Analysis for PCB Jet Printing using Ultra-Close Range Photogrammetry

Die Dissertation wurde am 15.11.2019 an der Fakultät für Informatik und Automatisierung der technischen Universität Ilmenau eingereicht.

Die Gutachten erstatteten:
Herr Univ.-Prof. Dr.-Ing. Daniel Baumgarten
Herr Univ.-Prof. Dr.-Ing. Jens Haueisen
Herr Univ.-Prof. Dr.-Ing. Hartmut Dickhaus

Die wissenschftliche Aussprache (öffentlicher Teil) erfolgte am 24.06.2020.

Name: M.Sc. David Heinemann
Address: Obere Karlstraße 1
91054 Erlangen

urn:nbn:de:gbv:ilm1-2020000175

„Wer die Geometrie begreift, vermag in dieser Welt alles zu verstehen.“

Galileo Galilei

Acknowledgement - Danksagung

Ich bedanke mich bei allen Freunden und Kollegen, die mich während der Bearbeitung meines Promotionsthemas und in der Anfertigung der Dissertation unterstützten. In erster Linie möchte ich einen besonderen Dank meinem Doktorvater Prof. Dr.-Ing. Daniel Baumgarten für die kompetente und verständnisvolle Betreuung der Arbeit aussprechen. Nicht weniger Dank gilt Herrn Prof. Dr.-Ing. habil. Jens Haueisen für die konstruktiven Anregungen und die Unterstützung bei Veröffentlichungen während der Promotionszeit. Er ermöglichte mir eine konstante Bearbeitung des Themas sowie ein stetiges Beschäftigungsverhältnis. Herrn Steffen Knabner danke ich für die Bereitstellung des Projektthemas, die stetige Gesprächsbereitschaft sowie für wichtige Hintergrundinformationen zum PCB Jet Printing und die Bereitstellung von Testdaten und Leiterplatten. Herrn Univ.-Prof. Dr. rer. nat. Gunther Notni sowie Herrn Dr.-Ing Andreas Breitbarth möchte ich meinen großen Dank für die Bereitstellung von Kalibrierkörpern und die Vermessung des in dieser Arbeit entstandenen Kalibrierkörpers aussprechen. Für die äußerst angenehme Zusammenarbeit und die hilfreichen Ratschläge sowie Kritiken und deren Einsatz für meine Weiterbeschäftigung am Institut bin ich besonders Frau Dr.-Ing. Dunja Jannek und Herrn apl. Prof. Dr.-Ing. habil. Andreas Keller verbunden. Ich danke Herrn Eberhardt Hamatscheck für die Bereitschaft zur Fertigung eines Kalibrierkörpers und Herrn Lorenz Esch für die wertvollen Hinweise in der Programmierung.

Ein ganz besonderer Dank gilt meinen Eltern Ina und Jörg sowie meinen Großeltern Brigitte, Siegfried, Marianne und Gerhard, ohne dessen Unterstützung mein Studium und damit der Weg zur Promotion nicht möglich gewesen wäre.

Für die Freundschaft und Hilfsbereitschaft bedanke ich mich in besonderem Maße bei Veikko Ruth, Marie Flucht, Christoph Evers und Nico Jobs, welche in anstrengenden und schwierigen Phasen der Promotion, aber insbesondere auch im Privatleben immer ein offenes Ohr und helfende Worte fanden.

David Heinemann
Ilmenau, 15. November 2019

Abstract

PCB Jet-Printing is an innovative method to deposit solder paste on printed circuit boards (PCB) without stencils. Through an ejector mechanism tiny droplets of solder paste are ejected onto the board. This new method allows for more flexible PCB layouts with smaller components. Due to the viscosity of the solder paste, air blisters can be trapped in the cartridge. This can lead to missing solder joints or deviations in the applied solder volume. Subsequent inspection systems in production lines are able to detect such errors and can ensure that only boards with the correct amount of solder paste are used for the further component placement. However, at this manufacturing step a correction of misprinted areas is not feasible with reasonable costs and effort. The objective of this thesis was the development of a quality assessment system in order to realize a built-in, real-time inspection of the printing process. In this respect, the solder joints volume should be measured before the PCB leaves the jet printer in order to correct the volume if necessary to minimize the uncertainties and thus increase the efficiency of the process. In the framework of this thesis, a photogrammetric reconstruction process including software and demonstrator system was designed, implemented and evaluated. A novel camera calibration method was proposed for the newly introduced field of *ultra-close range normal case photogrammetry* and was evaluated using the 2D reprojection error as well as the 3D reconstruction error. Well-known feature detectors and descriptors were employed and adapted to efficiently and reliably obtain image features. The methods are evaluated based on a newly introduced parameter, called *Solder Joint Feature Coverage*, that describes if a solder joint is properly covered by feature points. Based on that parameter the FAST detector is employed. The calibrated camera model was used in the processing step of image based feature extraction and model based matching correction in order to obtain true feature point correspondences. This represents a crucial challenge, since the examined solder joints show similar appearances, which makes a correct matching using feature descriptors more difficult. The corrected results of the feature matching step were assessed based on the recall and precision parameters as well as the reconstruction result. The evaluation shows the best results for a combination of the BRIEF descriptor and a hamming distance based feature matching with a subsequent camera model based matching correction. The solder joint areas in the images are detected by a texture based classification using the texture features of Haralick. Thus, only feature points of solder joints are processed. The adapted (motion corrected) camera models are used in combination with the corrected feature matches to reconstruct a 3D point cloud of the PCB. The point cloud is clustered into separate solder joints, where for each cluster a concave hull is computed. Based on the filtered hulls the volume of the solder joints is computed with the open source visualization toolkit (VTK). The processing steps were accelerated using NVIDIA's GPU acceleration CUDA and the Intel's threading building blocks (TBB) to run parts of the processing chain in parallel in order to reach on-line and real-time capability.

The resulting pipeline represents a suitable approach for camera calibration, solder paste - and feature detection. The successful implementation of these steps facilitates the point cloud reconstruction and subsequent volume computation. The calculated solder joint volumes are evaluated using a structured light based ground truth. The results show a suitable estimation with deviation of less than 12% for large solder joints with a volume of more than 30 nl, which coincides with the volume deviation of the printer. The overall deviation of the calculated volumes is 14%. The implemented real-time pipeline allows for these obtained solder joint volumes to be fed back to the printer to correct errors immediately during the printing process. Therewith, the overall efficiency and reliability of the printing process can be considerably enhanced.

Kurzfassung

Das PCB Jet Printing ist eine innovative Methode um Lotpaste ohne die Verwendung von Schablonen auf Leiterplatten (PCB - printed circtuit board) zu applizieren. Über eine Ausstoßvorrichtung eines Lotpastendruckers werden Lotpastentröpfchen an beliebige Positionen geschossen. Diese Methode erlaubt es flexiblere Leiterplattenlayouts mit kleineren Komponenten zu erstellen. Durch die Viskosität der Lotpaste können Luftblasen in der Kartusche eingeschlossen sein. Dieser Umstand kann zu Schwankungen in der applizierten Lotpastenmenge bis zu einem vollständigen Fehlen eines Lotpunktes führen. In Fertigungslinien enthaltene Inspektionssysteme sind in der Lage zu entscheiden, ob eine Leiterplatte korrekt bedruckt wurde und somit weiterverwendbar ist. Eine Korrektur fehlerhafter Platten ist in diesem Schritt mit vertretbaren Kosten und Aufwand jedoch nicht mehr möglich. Das Ziel dieser Arbeit ist daher die Entwicklung eines integrierbaren und online-fähigen Inspektionssystems um den Druckprozess zu überwachen. Dazu soll das Volumen der Lotpunkte ermittelt werden um dieses bei Bedarf zu korrigieren, bevor die Leiterplatte den Drucker verlässt. Somit können Unsicherheiten reduziert und die Effizienz des Systems erhöht werden. Für die Umsetzung wird eine photogrammertrische Rekonstruktion etabliert. Dies beinhaltet das Design und die Realisierung eines Demonstratorsystems sowie einer Software zur Datenerfassung, -prozessierung und Auswertung. Es wurde ein neuartiger Ansatz für die Kamerakalibrierung im neu eingeführten Bereich der *ultra-close range normal case photogrammetry* vorgeschlagen und über den 2D Reprojektionsfehler sowie den 3D Rekonstruktionsfehler evaluiert. Um effizient Bildmerkmale zu extrahieren wurden bekannte Merkmalsextraktoren und -deskriptoren untersucht und beispielsweise in Bezug auf eine Skalenraumprozessierung angepasst. Für die Evaluierung wurde ein neuer Parameter eingeführt (*Solder Joint Feature Coverage - SolFeC*), welcher die Abdeckung der Lotpunkte mit Bildmerkmalen beschreibt. Dieser Parameter bringt den FAST Detektor als geeignetes Verfahren hervor. Das kalibrierte Kameramodell wurde neben der Rekonstruktion zur modellbasierten Punktkorrespondenzkorrektur verwendet, da dieser Schritt im Nahbereich aufgrund des identischen Erscheinungsbildes verschiedener Lotpunkte eine hohe Herausforderung darstellt und eine Vektordistanz- oder Fundamentalmatrix- basierte Korrektur nicht ausreichend ist. Die Ergebnisse des feature matchings (Erstellen von Punktkorrespondenzen) wurde über die Parameter precision und recall sowie über das Rekonstruktionsergebnis bewertet. Daraus ergibt sich eine geeignete Kombination des BRIEF Deskriptors mit einem Hamming-Distanz basierten feature matching und anschließender Kameramodell-basierter Korrespondenzkorrektur. Die Lotpunktregionen in den Bilder werden über eine Textur-basierte Klassifikation mit Hilfe der Texturparameter nach Haralick detektiert. Somit werden nur Lotpunktmerkmale weiter verarbeitet. Mittels angepasster Kameramodelle wurden die Punktkorrespondenzen zur 3D Rekonstruktion einer Punktwolke verwendet. Diese wird in ein Clustering überführt wobei jedes Cluster einen Lotpunkt darstellt. Anschließend wird jede Lotpunktwolke über eine konkave Hülle vermascht und gefiltert um Ausreißer zu eliminieren. Aus den geschlossenen Hüllen kann mittels VTK (vi-

sualization toolkit) das Volumen bestimmt werden. Die einzelnen Prozessschritte wurden mittels NVIDIA's GPU Beschleunigung CUDA und Intel's Threading Building Block (TBB) parallelisiert um Onlinefähigkeit zu erreichen.
Die entwickelte Verarbeitungskette enthält geeignete Methoden zur Kamerakalibrierung, Lotpasten- und Merkmalsdetektion. Die erfolgreiche Implementierung dieser Schritte ermöglicht die Berechnung einer Lotpasten-Punkwolke sowie der entsprechenden Volumina. Diese werden mit Hilfe einer Referenz, welche durch eine Messung mittels strukturierter Beleuchtung erstellt wurde, evaluiert. Die Abweichung der berechneten Volumina innerhalb der Lotpunkte mit einem Volumen von mehr als 30 nl beträgt weniger als 12%, was mit der Schwankung des Lotpastendrucks korrespondiert. Die allgemeine Abweichung beträgt 14%. Durch die umgesetzte online-fähige Pipeline können Volumina als ein Feedback an den Drucker gesendet und auftretende Fehler direkt korrigiert werden. Somit kann die Effizienz und die Zuverlässigkeit des Druckprozesses wesentlich erhöht werden.

List of Abbreviations

2D two dimensional

3D three dimensional

ANN artificial neural network

AOI automatic optical inspection

API application programming interface

ASCII American Standard Code for Information Interchange

AXI automated X-ray inspection

BA bundleadjustment

BICF biologically inspired color features

BGA ball grid array

BRIEF Binary Robust Independent Elementary Features

BRISK Binary Rotation Invariant Scalable Keypoints

CAD computer aided design

CL Camera Link

CUDA Compute Unified Device Architecture

DoG difference of Gaussians

FAST Features from Accelerated Segment Test

FIFO first in first out

FLANN fast library for approximate nearest neighbors

fps frames Per second

GigE Gigabit Ethernet

GLCM gray level co-occurrence matrix

GUI graphical user interface

HoG histogram of oriented gradients

IC integrated circuit

MC multi-image calibration

MCCo multi-image calibration and correction

MSER Maximally Stable Extremal Regions

openCV open source computer vision

openMP open source Multi Processing

openGL open source Gaphics Library

ORB Oriented FAST and Rotated BRIEF

PCB printed circuit board

PCL Point Cloud Library

RANSAC random access sample consensus

ROI region of interest

SfM structure from motion

SIFT Scale Invariant Feature Transform

SMD surface mounted device

SolFeC Solder Joint Feature Coverage

SURF Speeded Up Robust Features

SC single-image calibration

SC2D single-image calibration - 2D Target

SC2DCo single-image calibration - 2D Target - compared to 3D corrected target

SCCo single-image calibration - corrected Target

SJI solder joint inspection

SPI solder paste inspection

SVD singular value decomposition

TBB Threading Building Blocks

VTK Visualization Toolkit

List of Figures

List of Tables

Contents

1 Introduction

1.1 PCB Manufacturing

Electronic devices become more powerful and more compact at the same time. A large amount of electronic components, like integrated circuits (ICs) and surface mounted devices (SMDs) at printed circuit boards (PCBs) are needed to ensure the efficiency of those devices. The miniaturization of electronic components has progressed significantly over the past years. This in turn leads to challenges when it comes to the manufacturing of PCBs and its subsequent quality assessment. Multiple efficient quality controls during PCB manufacturing are indispensable to ensure a high quality of PCB based devices. Figure 1.1 shows the classical process of the PCB manufacturing. Solder paste is applied to the PCB prior to the component placement. A specific amount of solder paste is needed, which depends on the size of the components, to establish a solid and conductive connection between the components and the board. Besides the precise positioning of the solder paste, the applied volume plays an important role [HLYK11]. A special significance is taken by the solder paste inspection (SPI), which belongs to the methods of automatic optical inspection (AOI). It examines the PCB for errors after the application of the solder paste (see Figure 1.1) and is the first quality control within the PCB manufacturing. Between 52% and 71% of errors during the PCB assembly can be lead back to the solder paste application [PTS+04]. Since the cost of a rejection increase tenfold by every manufacturing step the SPI is of high importance [Int18]. As it can be seen in figure 1.1 the SPI and other AOIs are integrated into the process chain, which is also called *online-capable* SPI (AOI). Therefore, an exceedingly short inspection time is desirable to avoid a decrease of the throughput. Not only the detection of the paste but also the volume measurement are essential tasks of SPIs. Another system to assess the quality of solder is the solder joint inspection (SJI). In general SPIand SJIs are able to detect the solder and the applied

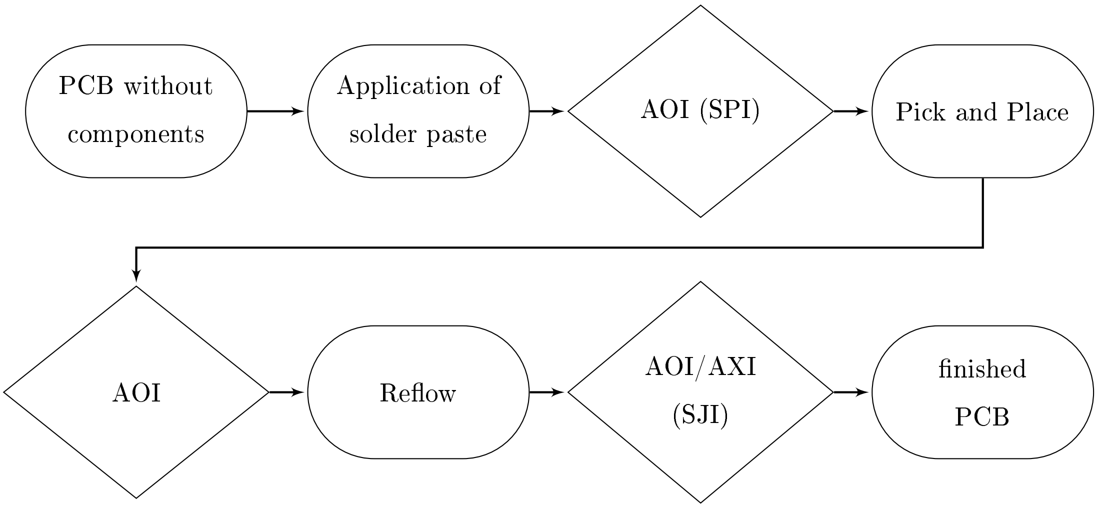


Figure 1.1 Flow chart of PCB manufacturing.

volume. However, the SJIs takes place after reflow soldering is done [OMR]. Reflow soldering is the process of heating the PCB to meld the solder within the paste and to activate the soft solder flux. The method of choice for applying solder paste was the screen printing using stencils, which has some disadvantages that are explained in figure 1.2. A novel method, the PCB jet printing overcomes these drawbacks.

1.2 PCB Jet Printing

1.2.1 General

For a long time the use of stencils was the state of the art for applying solder paste, which has some disadvantages in terms of separate manufacturing, delivery time, cleaning and storing. Since electronic components become smaller and PCB layouts more complex with a higher density of components (up to 100 components per cm^2), the use of stencils has become a challenging task, especially since smaller components need less amount of solder (thin stencil) than larger components (thick stencil) as it can be seen in figure 1.2. Stencils were identified to be the main cause for defects during the application of solder paste. PCB jet printing is a flexible method for the application of solder paste onto a PCB, which overcomes these problems with a shorter manufacturing time since there is no delay for ordering, manufacturing and delivery of stencils or the cleanup of used ones. Short term design changes are possible using the provided computer aided design (CAD) or gerber files, since the layout is given to the printing program [Coe2] [MYD07] .

1.2.2 Principle

The PCB jet printing is comparable to and inspired by the well known ink jet technology [Myd08]. The solder paste is stored inside a cartridge and is transferred via a stepper motor driven worm gear to the printhead (ejector), where a ceramic piezzo crystal exerts pressure to the paste channel. The piezzo based ejector mechanism deposits tiny droplets of solder paste onto the PCB (see figure 1.3) at a printing temperature of 30°C. The printing is done on the fly, which means the motion of the printhead is continuous while the solder paste is shot onto the board [Myd10a] [Myd10b].

1.2.3 PCB Jet Printer

PCB jet printing was professionalized by the Swedish company MyData with the commercially available jet printer "My500", which was used in this thesis. The principle of PCB jet printing and its production results form the basis of this work. A photograph of the My500 PCB jet printer and its inside can be seen in figure 1.5. The droplets have a volume of 5 to 15 nl, with a diameter of 330 - 470 µm. If a larger amount of volume is needed the printer shoots multiple droplets at the same position, with a speed of up to 200 droplets per second. At a volume of 5 nl the printer ensures a repeatability of 12 % and 8 % at 15 nl. The dimensions of printable boards range from 50 x 70 mm to 508 x 508 mm, with a maximum weight of 5 kg. The main parts of the printer are:

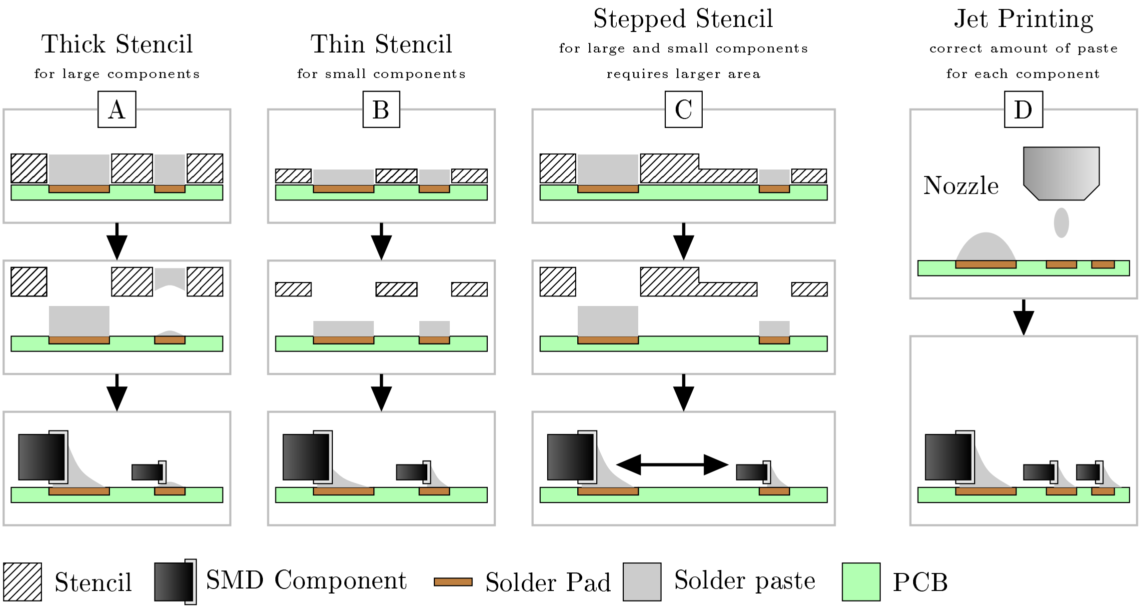


Figure 1.2 Comparison of stencil and jet printing - Drawbacks of screen printing (use of stencils): A-C top to bottom: Application of solder paste using a stencil; removing the stencil - solder paste should remain on the board; resulting solder joints at electronic components. D: application of solder paste using jet printing.
A: solder paste for small components may remain in thick stencils resulting. Large components obtain a good solder joint whereas lean joints may be present at small components. B: For small components thin stencil must be used to obtain good solder joints. This in turn leads to too little paste for larger components. C: The Problem can be solved by stepped stencils, which needs more space between small and large components. D: PCB Jet printing offers a better solution by printing any amount of solder paste at arbitrary positions [cf. Myd10b].

the positioning system, motion system, conveyors and the cassette, which includes the ejector and the cartridge, as the heart of the system. The housing consists of heavy stone epoxy composite material to ensure dimensional accuracy. An exemplary result of the PCB jet printing can be seen in figure 1.4. These test PCBs were used for examinations and developments in this thesis.

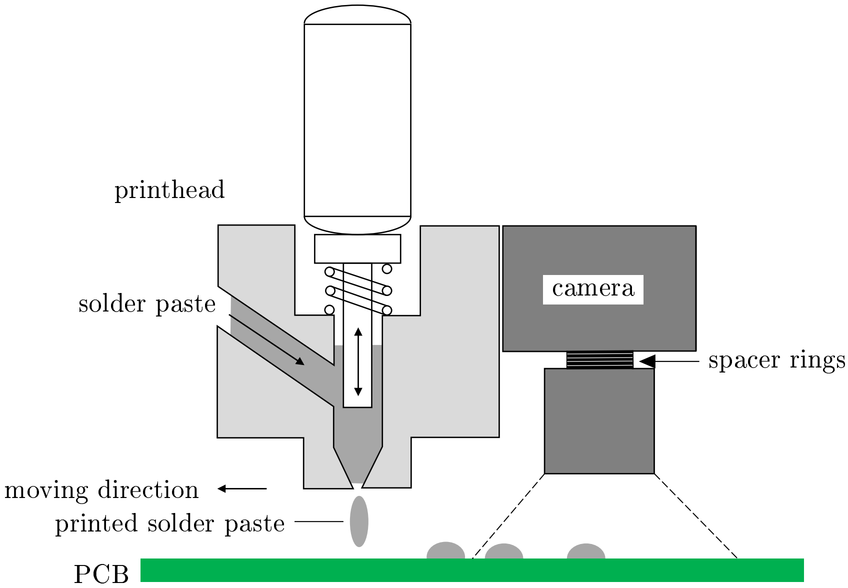


Figure 1.3 Schematic of printer head: solder paste is applied onto the PCB via a piezzo controlled ejector mechanism, a camera is attached to the printer head for the purpose of inspections. Spacer rings ensure a short working distance [adapted from Myd10b].

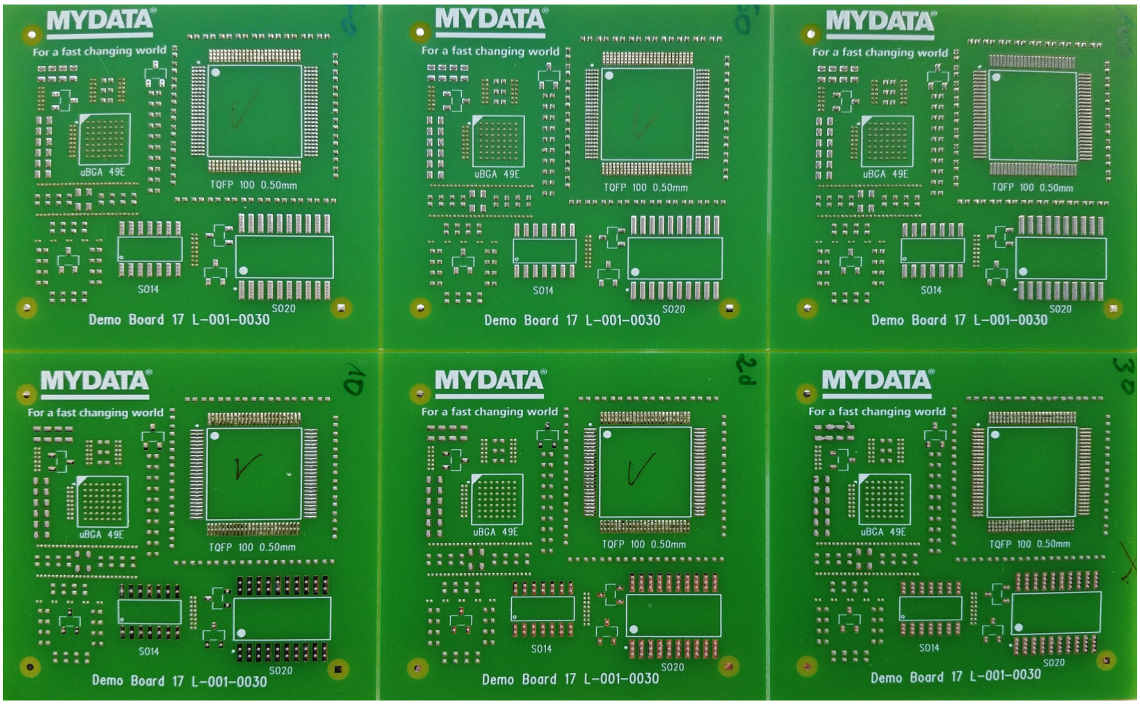


Figure 1.4 Photograph of an exemplary jet printed PCB, used for examinations and developments in this thesis. The 6 sub-PCBs contain varying amounts of solder paste.

1.2.4 Printing Problems

During jet printing, some errors in the printing result can occur. Due to the viscosity of the solder paste, air blisters can be trapped within the cartridge. The printer itself does not recognize if the

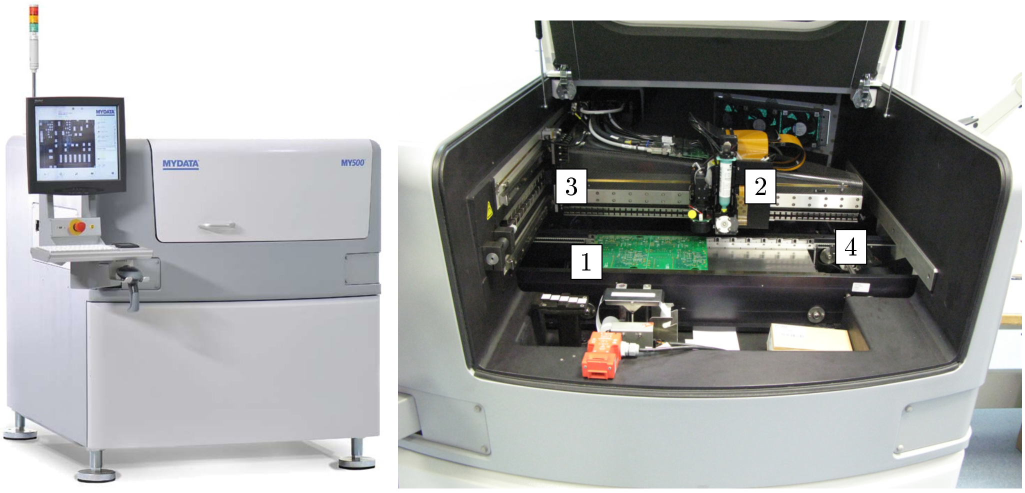


Figure 1.5 PCB Jet Printer: left: Photograph of the My500 jet printer [Myd10a]; right: Photograph of an opened jet printer: 1 - PCB, 2 - cassette, includes ejector and cartridge, 3 - T-shaped motion system, 4 - conveyors.

right amount of solder or partly air was printed. This in turn leads to deviations in the applied solder volume or in the worst case to a missing solder joint, which will result in a dry solder joint where no connection between board and component is established. In current manufacturing lines second devices, so-called AOIs, are used to detect printing errors and to reject misprinted boards after the actual printing process. However, this quality inspection is not able to correct the errors and can only give a posteriori statements if a board is printed correctly or has to be rejected.

1.3 Objectives

The application field of this thesis is at the interface between jet printing and solder paste quality assessment and tends to move the stand-alone SPI (AOI) into the printing process. The volumes of the printed solder joints need to be measured and returned to the printer as a feedback to overcome the mentioned issues of volume deviation and missing solder joints. Therefore, this thesis aims at an inspection system with the capability of online and built-in solder paste analysis to obtain the possibility of correcting misprinted areas before the PCB is leaving the printer. The terms *online* and *built-in* are used with different meanings in numerous application. In this thesis and the corresponding analysis, examinations and development, *built-in* considers the final setup with the ability to be integrated into the printer without high costs and effort. *Online* implies that all volumes are measured and potentially corrected prior to the ejection of the PCB from the printer. The PCB jet printer considered in this thesis has a solder joint volume repeatability of 12 % [Myd10a]. Thus, the maximum deviation of the inspection system with respect to detecting solder paste and estimating its volume is required to be below 12 %. Since the printer is used in clean and stable environments, the inspection system has to deal only with slight changes of the conditions (e.g. very small amount of dust at the PCB surface, changes in the ambient light) and has to be robust against these small variations. Different approaches with the capability of measuring 3D surfaces and volumes directly or within a proper pre-processing need to be examined in terms of their applicability for an integrated inspection system in order to reach the overall goal of solder joint detection and volume measurement. For the selected approach, a demonstrator system, taking into account on the conditions of the printer needs to be set up. Within this system, various algorithms and methods have to be realized, implemented and parameterized for an optimized processing.

In preliminary investigations, photogrammetry was identified as an auspicious solution for the overreaching objective. Therewith, multiple images of a solder joint can be used to obtain a 3D reconstruction via calibrated camera models. For that purpose, a software and hardware concept needs to be developed for the solder joint analysis, comprising the following components:

- hardware realization of a proper acquisition system (capable of integration)
- calibration
- triggered image acquisition
- image analysis for solder paste and feature detection
- 3D reconstruction and post processing
- volume estimation

Every step includes the investigation and analysis of state-of-the-art methods (which will be described in section 2) and the development of adapted or novel algorithms, in particular for camera calibration and image feature processing and analysis. Photogrammetry is used to reconstruct an object in 3D from at leas two images. Therefore, a moving camera or multiple spatially distributed cameras need to capture the object at different positions and viewing directions. The required 3D point reconstruction error should stay below 54 µm, which is the deposit accuracy of a single dot [Myd10a]. The following sub goals need to be realized to reach this overall aim. For an image based 3D reconstruction the used camera needs to be calibrated. Therefore, a calibration method that handles very short working distances and a fixed camera orientation - viewing direction (principal axis) normal to the PCB - needs to be established. For the reconstruction, image feature points need to be detected. Therefore, feature detectors and feature matching methods have to be examined and adapted to perform an efficient 3D reconstruction using the obtained camera models. The resulting point cloud needs to be post-processed in terms of clustering, surface reconstruction and volume estimation, where a proper method needs to be found and implemented. Finally, for the proof of principle and online capability of the developed processing chain, a laboratory demonstrator system including a software framework needs to be set up. Not a part of this work is the final implementation in the printer.

1.4 Considerations and General Approach

Multiple examinations and developments are needed to reach the overall goal of an inspection system with online and built-in capability. Since a photogrammetric pipeline (see figure 1.6) is pursued, a suitable imaging system needs to be set up (section 5). For a straightforward integration of the developed system into the printer, its geometric conditions and printing behavior needs to be examined and considered during the development. The technical implementation should not have any influence on the printing process, which means possible camera positions and illumination conditions have to be found. Inspecting the printer clarifies that the only position of a camera, without disturbing the printing process, is the printhead itself. Thus, the camera has only two degrees of freedom. The weight and dimensions of a camera play an important role when attaching it onto the printhead, since the additional weight must not influence the movement of the printhead. Due to the small size of the solder joints, the camera must be used with a short working distance. In this case, a working distance of 10-20 mm (depending on the used lens) is appropriate. When imaging objects within this working distance range, which can be reached by the use of spacer rings, the depth of field becomes very small. The stop of the lens needs to be closed as much as possible to enlarge the depth of field to maximum. This results in a strong decrease of brightness and requires additional illuminations. For a proper homogeneous lighting and to save weight at the printhead, bar lights at the edges of the overall printing area are considered to illuminate the PCB laterally. These illuminations enhance the solder joints structural appearance which makes the resulting images suitable for a novel texture based solder joint detection method. This approach is described in section 4.1. The camera captures images during the movement and overlapping image regions can be used for reconstruction and determination of position and volume of the solder joints. For this reconstruction, the camera needs to be calibrated which requires the development of a calibration method for the normal-case acquisition and the short working distance, which is described in section 3. For this special application field the term *ultra-close range normal case photogrammetry* is introduced. When a 3D point cloud is reconstructed, image feature points must be detected and matched across the images, which presupposes an examination of feature detectors and its parametrization to extract a proper amount of solder joint features (see section 4.2). For a suitable objective evaluation of the feature detector, a novel parameter, called *solder joint feature coverage - SolFeC*, is introduced to describe the coverage of solder joint image regions with features. Due to the high similarity of different solder joints and thus of the feature descriptors, a lot of false feature point correspondences are established during the feature matching. Since state of the art correction methods are not sufficient for this task, a correction method must be developed to use only correct feature matches between the images. The final development step of the processing pipeline is the 3D reconstruction in terms of point cloud generation and post-processing. At this step, all previous developments are combined and used for triangulation. Subsequently, a processing chain for filtering, surface reconstruction, clustering and measuring needs to be developed, which is a challenging task due to inhomogeneous point distributions of the point cloud. These steps will be presented in section 4.3. Another important consideration is the development environment.

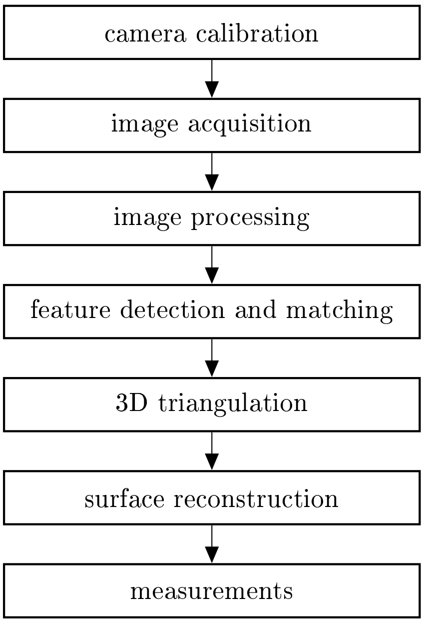


Figure 1.6 Detailed Photogrammetric Pipeline

The software of this thesis was developed using C++ under Linux, since the printer runs on this operating system. Finally, the algorithms are optimized to perform as fast and parallel as possible by introducing a specific programming pipeline concept called Intel Threading Building Blocks and acceleration techniques like NVIDIA's CUDA and OpenMP. During the development and implementation of the processing pipeline, a demonstrator system is developed to simulate the movement of the camera and to acquire images of the PCBs. The system will be employed to examine and proof single processing and pipeline sections, like camera calibration or feature extraction and to show the functionality of the complete system.

2 Basics and State of the Art

2.1 Solder Paste Inspection - Quality Assessment

2.1.1 Image based Analysis

Two dimensional inspection methods are SPI systems, which detect solder paste without the examination of the applied volume. The systems usually consist of a vision setup, including a monochrome or color camera and a powerful illumination, which varies in intensity and wavelength. Thus, the setups of such systems are trivial and cost effective.

Color Information Usually the solder differs from other components in the color information. This can be used to segment the PCB image within different color spaces to obtain only regions that contain solder. The acquired images are converted from the RGB color space into the HSV color space, where a saturation (S-channel) based threshold segmentation takes place. This is followed by a morphological filtering, where only the solder and the SMD components remain. The same operations are done within the value (V-channel), which gives the solder and the junction part of the SMD. After subtracting these results from each other and a final morphological filtering only the solder remains. With a known PCB position the deviations between required and applied solder position can be calculated. This can also be used as a pre-processing for further examinations, e.g. to run algorithms only in the extracted regions [SL12]. Another method tries to extract the solder paste based on the human perception. An RGB image is used to extract so called biologically inspired color features (BICF) from the color channels. Therefore, the channels red, green, blue and yellow (combination of red and green) are extracted and for each channel a 9-level image pyramid where the images are filtered with a Gaussian kernel and down sampled, is set up. From color and level differences 12 feature maps are obtained. Every map is divided into a 4×4 grid of regions and the mean value of each region is used to setup a 192 element vector for the image representation. This vector is used to classify good and poor solder joints by clustering. The method achieves a detection rate of 97% [JCT12].

Bayesian Statistics Bayesian statistics are based on dependent probabilities. In image processing *markov random fields* or a *marked point process* is used to detect structures in images. Markov random fields consider pixel values and a marked point process (MPP) considers previously detected object clusters as random variables and their occurrence at a certain position is expressed as a probability [BKJJ13], [DZ02]. A-priori knowledge can be included in the model which allows for multiple detection and classification tasks. Benedek et al. [BKJJ13] use a marked point process to detect solder paste and holes within a ball grid array (BGA), to avoid cracks when the chip is placed onto the solder paste when small holes are present. The detection of 50 - 120 solder joints takes about five to ten seconds, with a detection rate of 99.96% for solder paste and 92.40% for holes. An advantage of those systems is, that they can use a-priori information which makes it specific to single problems but errors that are not modeled will not be detected.

Artificial Neural Networks When using artificial neural networks (ANNs) for solder paste inspection, 2D images are used to compute image feature for the training of the network and subsequent classification [WZKL08]. Multi-channel light sources with different orientations can be used to obtain more image features, especially for 3D inspection. For the training also a ground truth is needed besides the extracted image features, which can be obtained by other inspection techniques like laser triangulation. For any change in terms of illumination, imaging and solder paste characteristics, the network needs to be trained again. After the training the network is able to estimate the solder paste volume. Features for the classifications are for example the area, shape and color gradient of solder paste areas[KYWY06].

2.1.2 3D Imaging based Analysis

Laser Line Detection and Triangulation For this method a laser beam is fanned out to a line and projected onto the PCB. A camera acquires an image of the line, whose shape and the known geometry of the setup (positions an orientations of PCB, laser and camera) is used to calculate the height profile along the laser line (see figure 2.1) [KM88] [LSZD07]. The variation of the height between solder paste and the PCB results in a shift of the laser line in the image. This shift is used to estimate the height of the solder paste at the laser line position [OKO+95]. A challenging task is the correct detection of the laser line, since artifacts like reflections, shadings

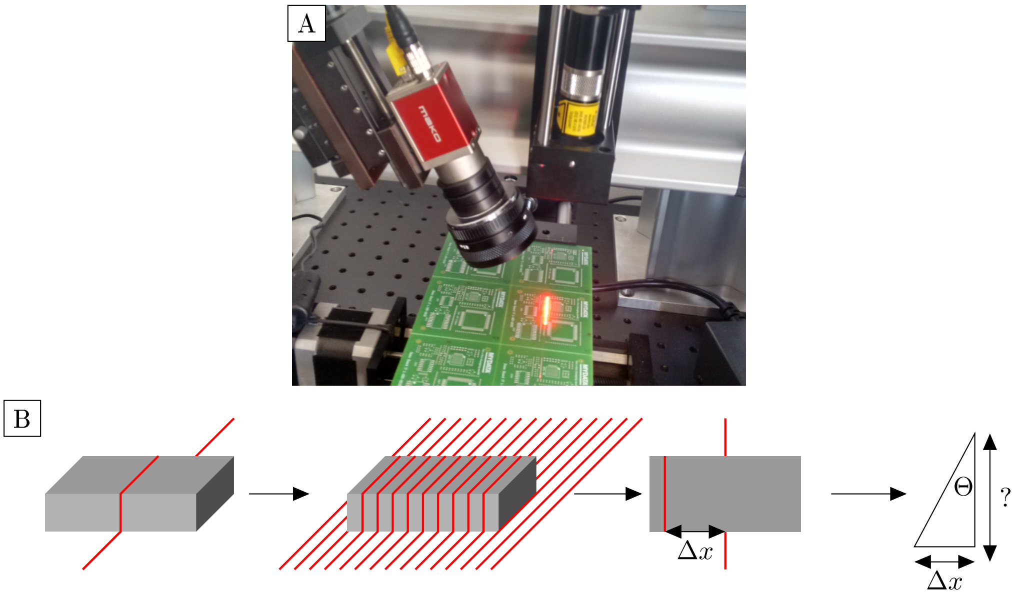


Figure 2.1 Laser Triangulation: A: Photograph of an laboratory laser line system; B: Principle of laser line detection. Due to the height caused laser line shift Δx and the known projection angle Θ the height of the current laser line position can be calculated [Bie11].

and the speckle noise of the laser lead to deviations in the intensity. Thus, the maximum of the intensity does not necessarily correspond to the center of the line [UMG12]. The speed of the method can be increased by using multiple lines which are color coded for spacial assignments [BK87] or temporal multiplexed [SPB04]. A movement of the object or the SPI along the laser line's perpendicular direction needs to be performed for a complete examination of the object [HvACL96].

Color Coding In color coding inspection systems different light sources with varying colors are used to examine orientation [PC09] and height [LL99] of the solder paste. Figure 2.2 shows an exemplary system. The systems contains of red, blue and white illuminations. For each active light source an image is taken by a camera. In the combined image planar areas appear violet, while ascending and descending areas are colored by the opposite light source. From the color based segmentation, statements on the structure of the solder paste can be given.

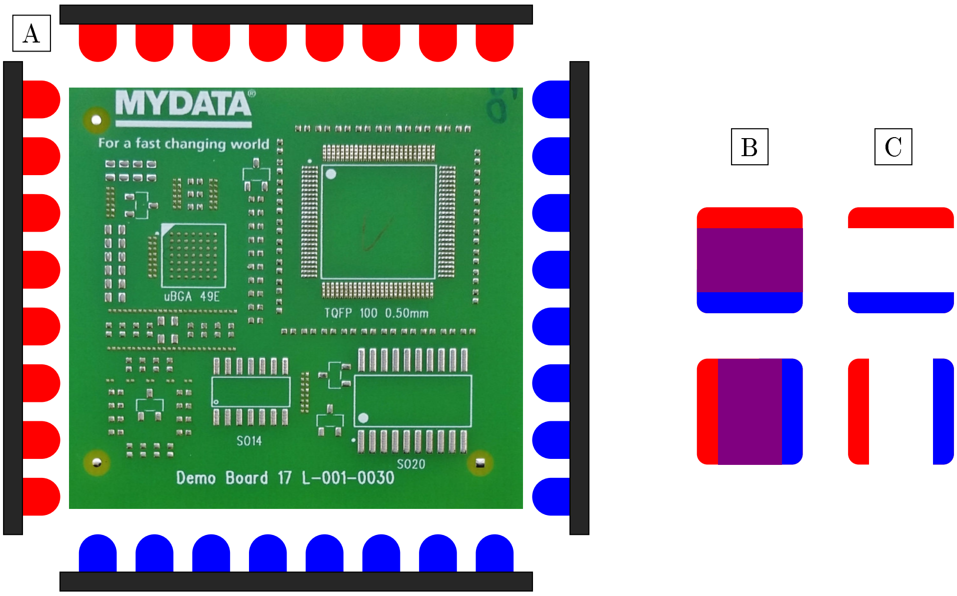


Figure 2.2 Color based Inspection: A: Laboratory System for color coded PCB inspection; B: input images - increasing height of the solder paste reflects the opposite light color (red and blue). A plane surface is represented by the combination of both illuminations (purple area) C: Segmentation of illuminated areas - the height change can be extracted by the color segmentation of red and blue [PC09].

Moiré Analysis The examination of moiré fringes allows for a direct capture of the height profile of an object. These fringes arise from the overlap of two identical gratings which are twisted against each other by an angle α. This angle and the raster period of the two gratings define the period of the resulting moiré fringe [BFH01]. Within the technical realization the first grating is projected via a light source and a projection lens onto the examination surface. The second grating in combination with a viewing lens is placed in front of a camera. The combination of projection and viewing grating results in the moiré fringes. Multiple acquisitions with a phase shift of the grating is done to remove the viewing grating from the image by considering the mean value of the images. From these resulting intensity distributions the phase of the pattern is calculated. Together with the wave length the height profile of the surface is obtained [CK98]. A sketch of this principle is depicted in figure 2.3.

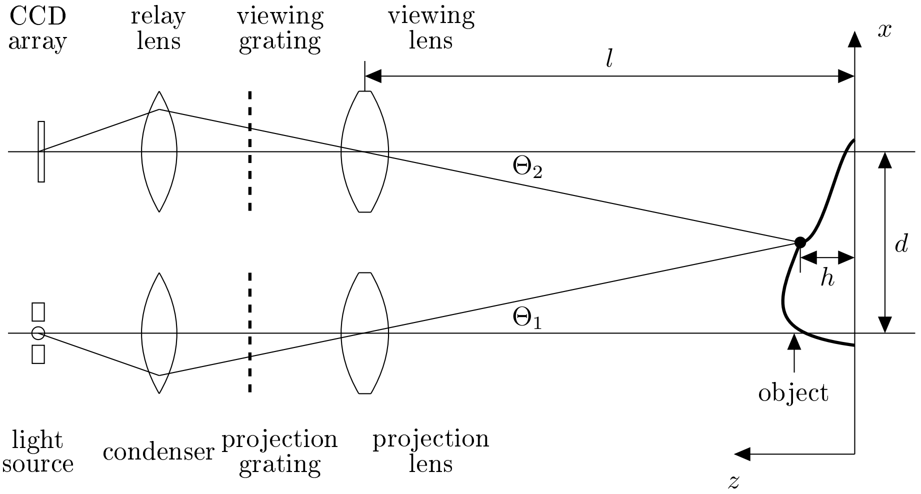


Figure 2.3 Schematic of moiré based height profile estimation: light rays are passing a condenser are structured by a projection grading. The grating is than projected via a projection lens onto the object. A structure leaves the lens under the angle θ_1. Depending on the height of the object the ray hits the viewing grating via the viewing lens under angle θ_2 and is projected onto the CCD Array [CK98].

Fourier Transform Profilometry and Phase Measurement Profilometry Similar to the moiré analysis a periodic pattern is projected onto the surface as well as onto a reference plane via a projector and is acquired by a camera. Due to the surface topology the camera acquires a deformed grating image. The pattern is processed in the special frequency domain and in the space domain. Therefore, 1D Fourier transform is used to filter the pattern in a way, that only the fundamental component remains. Afterwards the inverse Fourier transform is applied. The height distribution varies the phase change[TM83]. Using the phase change together with the optical geometry, the height profile can be calculated. [SC01]

Stereo Vision Nakazawa et al. [NS14] and Li et al. [LBKP16] describe SPI-Methods based on stereo vision system. Two cameras, that are equipped with a ring light acquire a single solder joint (see figure 2.4). The cameras are oriented in a way, that each camera acquires a reflection at the highest point of the solder joint. The reflection is used as a feature for the subsequent reconstruction.

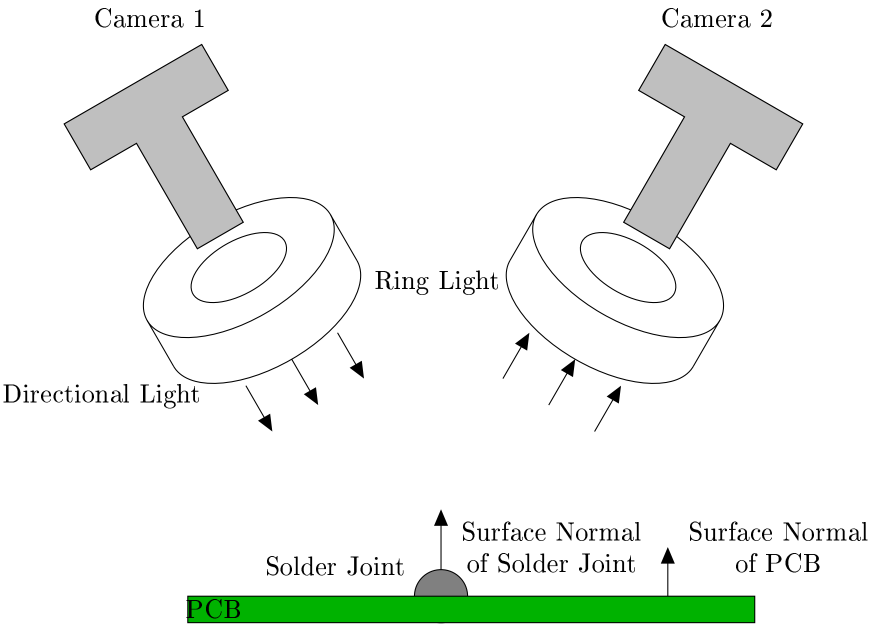


Figure 2.4 Schematic of stereo vision based SPI - ring light creates reflection points that are viewed by the second camera to generate features for reconstruction [LBKP16].

Shape from Shading Shape from Shading is a method to obtain the 3D shape of an object from its shading, which results from different illumination directions. Based on that shading the surface normals are calculated, since the normal has an influence on the strength and direction of the reflected light. By using an illumination with known structure and position, the estimation of the surface normals can be defined as an inverse problem. Starting with a unique distribution of point coordinates with zero height ($z = 0$) an iterative loop estimated the height of each point based on its pixel intensity. Due to varying reflection properties of different PCB components the position of the solder paste needs to be determined, which can be done with an image based detection or by the use of the PCB's design data (gerber files). Figure 2.5 shows a sketch of this technique [Kon08].

Automated X-Ray Inspection In automated X-ray inspection (AXI) an X-ray image of the PCB is captured. This method is of high importance when PCB components don't allow a direct view onto the solder paste for example at a BGA where the chip is placed over multiple solder joints. In this field of application many techniques are available. Besides the simple 2D X-ray inspection computed tomography is used to obtain a 3D reconstruction of solder paste structures. This allows for the inspection of the solder paste within multiple cross sections. Thus, defects, which are not visible from the outside (e.g. small cracks) can be detected as visualized in figure 2.6. The AXI has a high detection rate of errors but needs additional X-ray protection and has high costs [TMTF07].

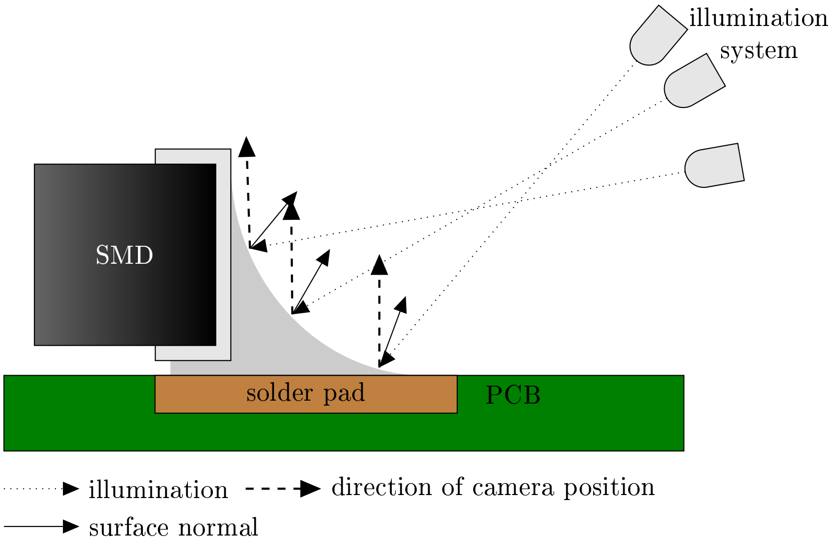


Figure 2.5 SJI using shape from shading. Depending on the position of the illumination the solder joints shade is changed. The change of the shading is interpreted for height estimation [KCMP99].

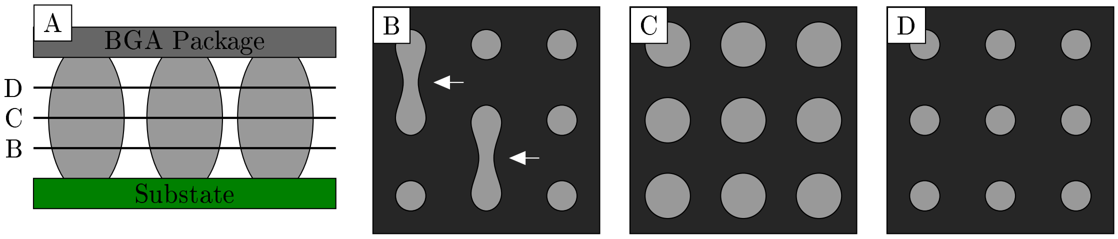


Figure 2.6 AXI of an BGA A: positions of examined slices; B-D: sketch X-ray images of slices B-D. Defects are visible at slice B (arrows) which are not visible from outside [TMTF07].

2.1.3 Comparison

The presented inspection methods can be divided in methods with and without 3D information like volume and height. Methods without a 3D measurement can be evaluated by the detection rate of solder paste and/or errors (see table 2.1). Especially statistical methods are able to detect the solder paste with a detection precision of over 99%. However, the results of the 2D methods rely on a-priori knowledge of the appearance of the solder paste. If that appearance differs from the previously assumed properties, the detection will fail. The precision of 3D inspection methods depends on the 3D reconstruction of the algorithms. Usually the methods are evaluated by measuring a precisely manufactured calibration normal. The resolutions of the mentioned methods are listed in table 2.2. Laser and Moiré based methods deliver the best resolution, whereas Fourier transform profilometry and phase measurement profilometry, stereo vision and artificial neural networks result in resolution less than 25 μm. However, these values are considered as examples for the corresponding applications to reveal its abilities and show a inter-study and also

Table 2.1 Comparison of solder paste inspection methods with 2D information by detection precision

Method	Detection precision
BICF	97% error detection [JCT12]
MPP	99,9% solder paste detection [BKJJ13]
Color coding	97,7% solder paste detection [PC09]

Table 2.2 Comparison of solder paste inspection methods with 3D information by resolution

Method	Resolution
Laser based	1,35μm [LSZD07]
Moiré	2.2 μm [KCO99]
FTP/ PMP	25.38 μm [HP08]
Stereo vision	25.15 μm [LBKP16]
ANN	39.9 μm [KYWY06]

an inter-manufacturer variety.

Each of the presented inspection methods has its advantages and drawbacks. However, they have in general, that the PCB or solder paste is inspected after the manufacturing, the application of solder paste is finished, respectively. Due to the acquisition procedures which require a scan-like motion or a non moving object, these methods are not suitable for an built-in solution for PCB jet-printing analysis. Another reason is the imaging setup of these methods. Especially X-ray based inspections require a specific housing of the system including X-ray protection.

2.2 Camera Basics

2.2.1 Intrinsic and Extrinsic Camera Parameters

A geometric camera model describes how a 3D world point $\boldsymbol{X} = [X, Y, Z, 1]$ is projected or transformed onto the image plane (sensor plane) to the 2D image point $\boldsymbol{x} = [u, v, w]^T$ ($x = u/w$, $y = v/w$) via a transformation P. Every camera model consists of the parameters that determine the position, orientation and the geometric imaging properties of the camera. These parameters are divided into two groups: the intrinsic (interior) and extrinsic (exterior) camera parameters. The extrinsics describe the translation of the cameras projection center C via a translation vector $\vec{t}$ within a 3D world coordinate system and the rotation of the camera or more accurate of the camera coordinate system (x_c, y_c, z_c), where x_c and y_c are parallel to the image coordinate system and z_c corresponds to the cameras principal axis or principal ray, respectively. The intrinsic parameters are the principal distance c_x and c_y (usually expressed in pixels), which are not equal for example when pixels with non unique dimensions are used, the principal point (h_x, h_y), where the principal ray enters the cameras sensor (this is not necessarily the center of the sensor) and the shear factor

s [HZ03] [LRKB13].

The most relevant camera model with these parameters is the pinhole model, which is also known as the projective camera. This model is used for the photogrammetric reconstruction in this thesis and will be explained in detail. An adaption of this model is the parametric model of a microscope proposed by [ZN99]. The imaging setup where the model is used is similar to the setup of this work, where a camera with a short working distance is pointing normal onto the examination area (PCB). Thus, this model is examined for applicability and is also described in the following sections.

Other models are known as "cameras at infinity" (with affine and non-affine cameras), push-broom cameras and line cameras. However, these cameras do not matter for this work and will not be explained. Details on these models can be found in [HZ03].

2.2.2 Pinhole Model

The model of a pinhole camera defines a central perspective projection of a world point $\boldsymbol{X}$ to an image points $\boldsymbol{x}$ [FW16]. For each world point only one ray enters the optical center and hits the sensor plane, resulting in every world point to be in focus [BK08, p. 371]. The principal distance c and working distance Z determines the magnification m:

$$m = \frac{Z}{c} = \frac{\boldsymbol{X}}{\boldsymbol{x}} \tag{2.1}$$

According to figure 2.7 for the x-coordinate of an image point $\boldsymbol{x}$ applies:

$$\frac{-x}{c} = \frac{X}{Z} \text{ and with } m\text{:} \quad -xm = X \tag{2.2}$$

For mathematical simplification the image plane is flipped around the optical center C, which results in figure 2.8. The optical center, also called camera center, is the origin of the camera

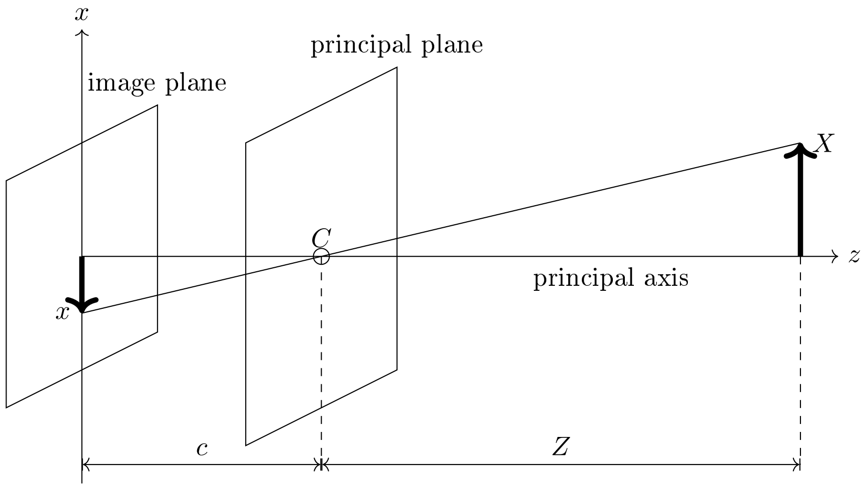


Figure 2.7 Pinhole camera model - Z: working distance, c: principal distance; C camera center; X: x-coordinate of world point $\boldsymbol{X}$; x: image coordinate of image point $\boldsymbol{x}$ [HZ03] p.154.

coordinate system. The cameras z-axis is defined as the principal axis and is perpendicular to the image plane. The projection can now be written as:

$$\left(\begin{array}{c} X, Y, Z \end{array}\right)^T \mapsto \left(\begin{array}{c} f\frac{X}{Z}, f\frac{Y}{Z} \end{array}\right)^T \tag{2.3}$$

Homogeneous coordinates are introduced to express the projection as a matrix operation. Thus, the world points and the image points are represented by homogeneous vectors and the projection is expressed as a mapping between their coordinates including the cameras principal distance:

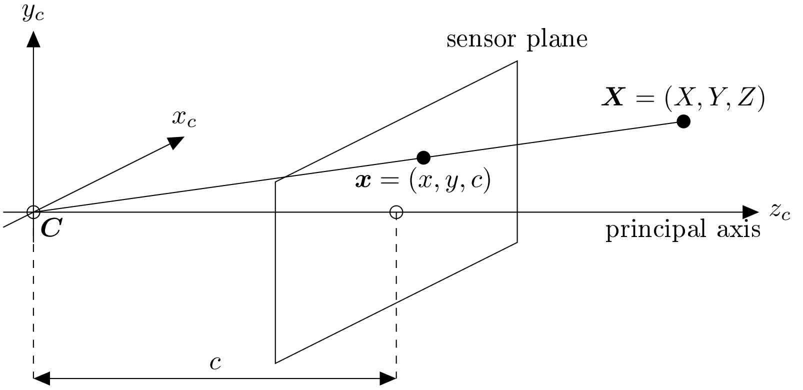


Figure 2.8 Mathematical simplification of the pinhole model [HZ03] p.154.

$$\begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \mapsto \begin{pmatrix} cX \\ cY \\ Z \end{pmatrix} = \begin{bmatrix} c & & & 0 \\ & c & & 0 \\ & & 1 & 0 \end{bmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \tag{2.4}$$

The homogeneous perspective projection matrix P consists of all transformations and camera parameters (see figure 2.9) that are necessary to map the world point to the image point.

$$\boldsymbol{x} = P\boldsymbol{X} \tag{2.5}$$

In practice the origin of the image coordinate system does not coincide with the principal point (h_x, h_y). Therefore, an offset has to be added to the general mapping:

$$\left(\begin{array}{c} X, Y, Z \end{array}\right)^T \mapsto \left(\begin{array}{c} c\frac{X}{Z} + h_x, c\frac{Y}{Z} + h_y \end{array}\right)^T \tag{2.6}$$

This is rewritten in homogeneous coordinates:

$$\begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \mapsto \begin{pmatrix} cX + Zh_x \\ cY + Zh_y \\ Z \end{pmatrix} = \begin{bmatrix} c & & h_x & 0 \\ & c & h_y & 0 \\ & & 1 & 0 \end{bmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \tag{2.7}$$

where the camera calibration matrix K is introduced:

$$K = \begin{bmatrix} c & & h_x \\ & c & h_y \\ & & 1 \end{bmatrix} \tag{2.8}$$

A further parameter, the shear factor s, is added to the calibration matrix K. This factor is zero for usual cases and becomes non-zero if there is a skewing of the pixel elements in the sensor array, which results in non-perpendicular x- and y- axis. For non-quadratic pixels different values for the principal distance along x- and y- direction occur.

$$K = \begin{bmatrix} c_x & s & h_x \\ & c_y & h_y \\ & & 1 \end{bmatrix} \tag{2.9}$$

In general, world points are not expressed in camera coordinates, but within the world coordinate system. However, the previous equations consider the world points to be in camera coordinates. The camera coordinate system has to be transformed into the world coordinate system by rotation matrix R and a translation vector t, as visualized in figure 2.9. A world point $\boldsymbol{X}$ is transformed to camera coordinates $\boldsymbol{X}_{Cam}$ by equation 2.10:

$$\boldsymbol{X}_{cam} = R\boldsymbol{X} + t \tag{2.10}$$

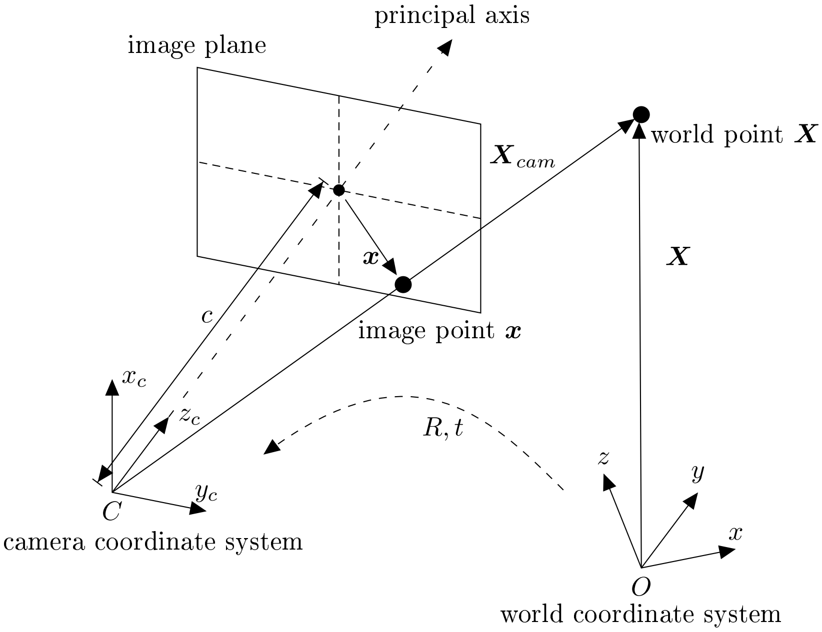


Figure 2.9 Pinhole model and transformations - R: rotation matrix, t:translation vector, to transform between world coordinate system (x, y, z) and camera coordinate system (x_c, y_c, z_c)[HZ03, p.156]

The projection matrix P is then compound as:

$$P = K \cdot R\,[I|t] \tag{2.11}$$

2.2.3 Optical Microscope Model

The imaging setup of the practical work of this thesis is similar to the microscope scheme proposed by Zhou et al. [ZN99]. This model was based on the pinhole model, where some changes and additions have been made to match the characteristics of an optical microscope. As it can be seen in figure 2.10 an optical tube T_{op} is added to the principal distance and the distance d between the object plane and the focal plane is introduced. T_{op} corresponds to the spacer rings that are used within the demonstrator setup described in chapter 5. The image plane is nearly parallel to the object plane. The mentioned additional parameters belong to the intrinsics, whereas extrinsics remain the same compared to the pinhole model. From figure 2.10 the projection becomes:

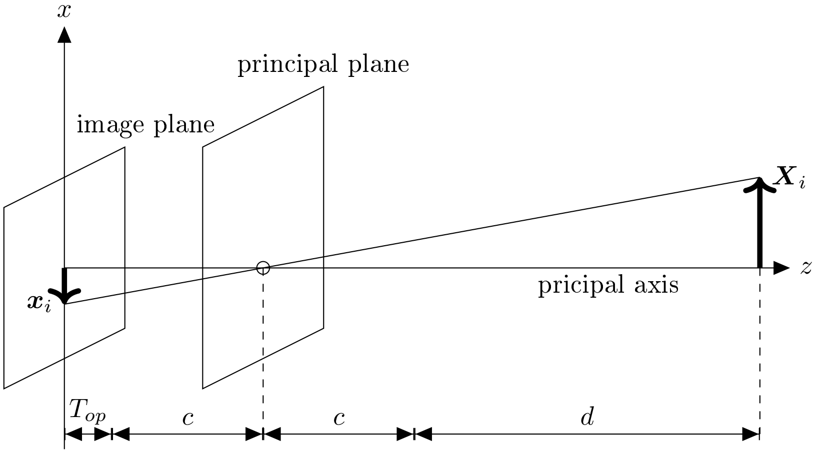


Figure 2.10 Parametric microscope model, adopted to the inspection system: X_i: i-th world point; x_i: i-th image point, T_{op}: spacer ring size (optical extension tube); c: principal distance; d: distance between object and focal plane [ZN99].

$$\left(\; X, Y, Z \; \right)^T \mapsto \left(\; (T_{op} + c) \, \tfrac{X}{Z}, (T_{op} + c) \, \tfrac{Y}{Z} \; \right)^T \tag{2.12}$$

The world points $\boldsymbol{X}_i = [X, Y, Z, 1]^T$ are transformed to image coordinates $\boldsymbol{x}_i = [x, y, 1]^T$ for the parallel case via:

$$\begin{aligned} x &= (T_{op} + c) \, \frac{r_{11}X + r_{12}Y + t_x}{r_{31}X + r_{32}Y + f + d} \\ y &= (T_{op} + c) \, \frac{r_{21}X + r_{22}Y + t_y}{r_{31}X + r_{32}Y + f + d} \end{aligned} \tag{2.13}$$

The elements of the calibration matrix are then:

$$\alpha_x = (T_{op} + c) \, m_x; \;\; \alpha_y = (T_{op} + c) \, m_y \tag{2.14}$$

where m_x and m_y are scale factors for non-quadratic pixels. With the assumption that $s = 0$ and the center of the image is a sufficiently accurate estimation of the principal point [Hor00], the calibration matrix becomes:

$$K = \begin{bmatrix} \alpha_x & 0 & x_0 \\ 0 & \alpha_y & y_0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.15}$$

2.2.4 Distortions and Lens Model

Every camera uses lenses to project the world points onto the sensor. In real imaging applications the perspective central projection is distorted by non-linear lens properties [HZ03, p. 189]. These are modeled by polynomial functions to remove lens distortion effects from images. By inverting these functions they can also be used to add a synthetic distortion to an ideal projection. The distortions consist of tangential and radial distortion components. In general a distinction is made between two different radial distortions, which depends on the position of the aperture. If the aperture is in front of the lens, one obtains a barrel distortion and a pincushion distortion if the aperture is placed behind the lens. The effects can be seen in figure 2.12. Due to these distortions, points at the edge of the image show a larger deviation from their real position than points at the center, since rays are refracted more the farther they are from the center of the lens [BK08, p. 375]. As it can be seen in figure 2.11 the projected image of the world point $\boldsymbol{X} = (X, Y, Z)^T$ is in front of the image plane. Due to the lens properties, a point is not imaged to a point but to a spot, which is called the circle of confusion or blur circle u. If the diameter of the circle of confusion is smaller than one pixel the image of the object point appears sharp. Thus, not only a single point but all points with a certain distance, the so called depth of field, are imaged sharp on the sensor. In turn the depth of field defines the distance T from the furthest to the nearest world point that are in focus (see equations 2.16 to 2.19). Let R define the distance from the lens to the furthest point and S the distance to the nearest Point. The distance to the 3D point with a projected image exactly on the image plane is denoted as a.

$$T = S - R \left| R = \frac{a}{1+K}, \; S = \frac{a}{1-K} \right. \tag{2.16}$$

where

$$K = \frac{k(a-f)u}{f^2} \tag{2.17}$$

and

- a = exactly focused world point
- f = focal length
- k = f-number

The circle of confusion u is then calculated as follows.

$$u = \frac{S-R}{S+R}\frac{f^2}{k(a-f)} \tag{2.18}$$

The depth of field is defined as:

$$T = S - R = \frac{2uk(1+\beta')}{\beta'^2 - \left(\frac{uk}{f}\right)^2} \tag{2.19}$$

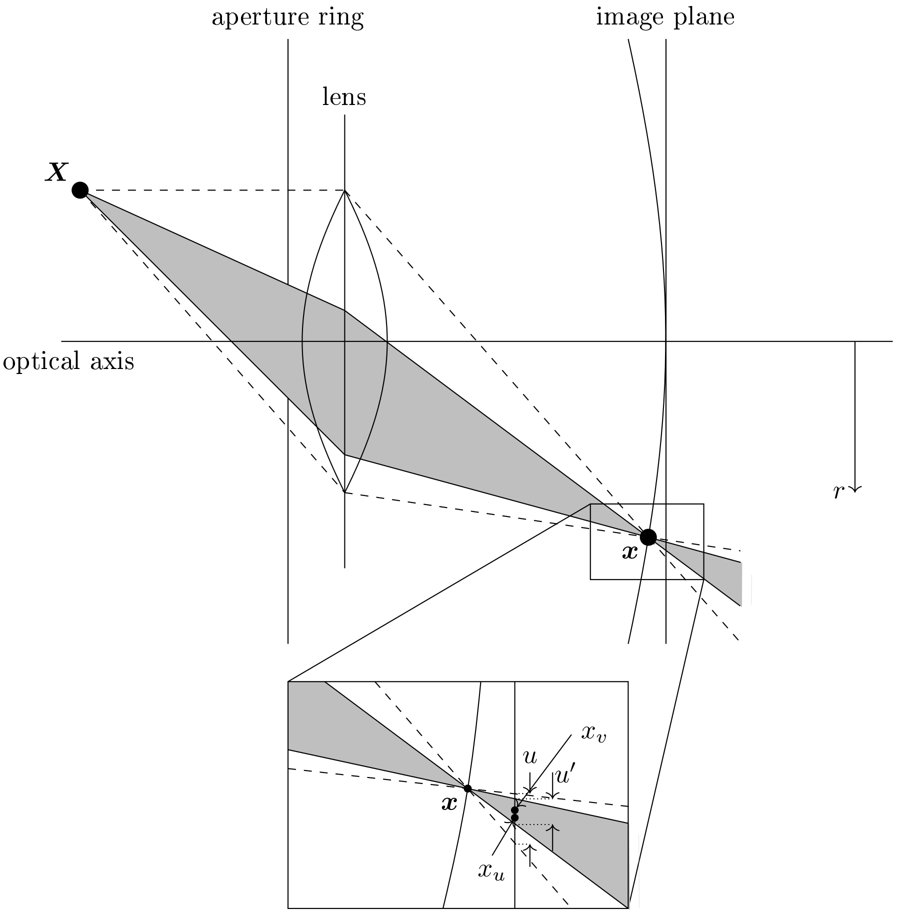


Figure 2.11 Visualization of the radial distortion. The projected image $\boldsymbol{x}$ of $\boldsymbol{X}$ (dashed lines)lies in front of the image plane and creates a circle of confusion u. A smaller lens aperture can reduce the effect (gray region), which increases with the radius r from the image center. The center of the circle confusion x_u then moves to x_v. The new circle of confusion is determined by u' (cf. [TBW03, p. 2]).

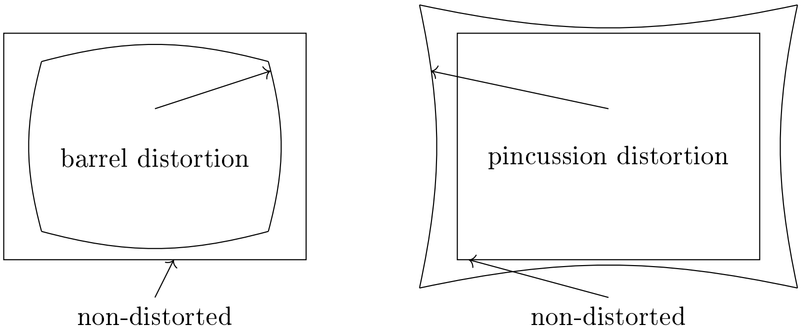


Figure 2.12 Sketch of radial distortions: left: barrel distortion, right: pincushion distortion (cf. [TBW03, p. 2]).

where *beta'* is the imaging scale [LRKH11]. A smaller lens aperture can be used to reduce the circle of confusion and to enlarge the depth of field, which results in a sharper image. This also moves the center of blur circle from x_u to x_v as it can be seen in figure 2.11.

The radial distortion coefficients k_i in equation 2.20 increase as the focal length decreases [RVSS10] [Hug10] [KB16]. The radial effect is modeled by the following polynomial function, which is aborted after the 4th term:

$$\begin{aligned} x_{cor} &= \Delta x_d \left(1 + k_1 r_d^2 + k_2 r_d^4 + k_3 r_d^6 + k_4 r_d^8\right) \\ y_{cor} &= \Delta y_d \left(1 + k_1 r_d^2 + k_2 r_d^4 + k_3 r_d^6 + k_4 r_d^8\right) \end{aligned} \tag{2.20}$$

with $\Delta x_d = (x_d - h_x)$, $\Delta y_d = (y_d - h_y)$, $r = \sqrt{(\Delta x_d)^2 + (\Delta y_d)^2}$; x_d is the distorted pixel and h_x is the principal point.

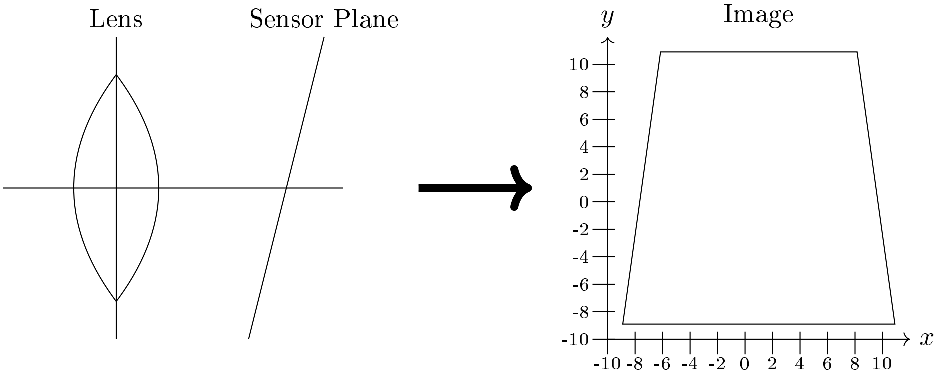


Figure 2.13 Sketch of the tangential distortion: left: mechanic cause, right: effect of tangential distortion (cf. [BK08, p. 377])

The tangential distortion, also called decentering distortion [Bro66]; caused by improper alignment between the lens and the sensor (see figure 2.13), is defined by the coefficients t_i:

$$\begin{aligned} \delta_x &= t_1 \left(3\Delta x_d^2 + \Delta y_d^2\right) + 2t_2 \Delta x_d \Delta y_d \\ \delta_y &= 2t_1 \Delta x_d \Delta y_d + t_2 \left(\Delta x_d^2 + 3\Delta y_d^2\right) \end{aligned} \tag{2.21}$$

Thus, the compounded model is:

$$\begin{aligned} x_{cor} = \Delta x_d &\left(1 + k_1 r_d^2 + k_2 r_d^4 + k_3 r_d^6 + k_4 r_d^8\right) + \\ &t_1 \left(3\Delta x_d^2 + \Delta y_d^2\right) + 2t_2 \Delta x_d \Delta y_d \\ y_{cor} = \Delta y_d &\left(1 + k_1 r_d^2 + k_2 r_d^4 + k_3 r_d^6 + k_4 r_d^8\right) + \\ &2t_1 \Delta x_d \Delta y_d + t_2 \left(\Delta x_d^2 + 3\Delta y_d^2\right) \end{aligned} \tag{2.22}$$

2.3 Camera Calibration

2.3.1 Tsai's Camera Calibration

The camera calibration method of Robert Tsai [Tsa87] consists of two steps. In the first part the rotation matrix R as well as the x- and y-components of the translation vector t are determined. Therefore, a calibration target with known geometry is acquired. When using a planar calibration target the z-coordinate of the markers is set to zero. Then, the pixel coordinates of the calibration markers are estimated, where a sub-pixel localization enhances the accuracy of calibration. Each detected marker in the image is assigned to its 3D world coordinate. Thus, the set of point correspondences $\boldsymbol{x}_i \leftrightarrow \boldsymbol{X}_i$ is established. For every point $\boldsymbol{X}_i = (X_i, Y_i, Z_i)$ and the corresponding point $\boldsymbol{x}_i = (x_i, y_i)$ as well as $x_i' = x_i - x_0$ and $y_i' = y_i - y_0$ the following linear equation is set up:

$$\begin{bmatrix} y_i'X_i & y_i'Y_i & y_i' & -x_i'X_i & -x_i'Y_i \end{bmatrix} \begin{bmatrix} t_y^{-1}r_1 \\ t_y^{-1}r_2 \\ t_y^{-1}t_x \\ t_y^{-1}r_4 \\ t_y^{-1}r_5 \end{bmatrix} = x_i' \tag{2.23}$$

In this equation r are the elements of the rotation matrix and t_x, t_y are the components of the translation vector. With the number of points $N >> 5$, an over-determined system of equations is established. The system can be resolved for the unknowns $t_y^{-1}r_1, t_y^{-1}r_2, t_y^{-1}t_x, t_y^{-1}r_4, t_y^{-1}r_5$. These unknowns set up the solution vector, which is denoted as SV and represents an interim solution. The final parameters $t_y, t_x, r_{11}, r_{12}, r_{21}, r_{22}$ are obtained from SV as it is shown in [Tsa87, p. 331 – 332], which results in the equations in 2.25.

$$t_y^2 = \frac{S_r - \sqrt{S_r^2 - 4\left(SV_1SV_5 - SV_4SV_2\right)^2}}{2\left(SV_1SV_5 - SV_4SV_2\right)^2} \mid S_r = SV_1^2 + SV_2^1 + SV_4^2 + SV_5^2 \tag{2.24}$$

and

$$\begin{aligned} r_{11} &= SV_1 \cdot t_y \\ r_{12} &= SV_2 \cdot t_y \\ t_x &= SV_3 \cdot t_y \\ r_{21} &= SV_4 \cdot t_y \\ r_{22} &= SV_5 \cdot t_y \end{aligned} \tag{2.25}$$

Since $|t_y|$ has two solutions (negative or positive), one has to determine its sign. Therefore, an arbitrary object point $\boldsymbol{X}$ with projected image coordinates outside the image center is transformed (using a positive t_y) via:

$$x = r_{11}X + r_{12}Y + t_x \ ; \ y = r_{21}Y + r_{22}Y + t_y \tag{2.26}$$

If the sign of the x- and y- components of the world points are the same as their corresponding image points, the sign of t_y stays $+1$, otherwise -1 is used. The other elements of R result from [Tsa87, S. 331]:

$$R = \begin{bmatrix} r_{11} & r_{12} & \sqrt{1 - r_{11}^2 - r_{12}^2} \\ r_{21} & r_{22} & s\sqrt{1 - r_{21}^2 - r_{22}^2} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \tag{2.27}$$

where $s = sign(r_{11}r_{21} + r_{12}r_{22})$ and the elements (r_{31}, r_{32}, r_{33}) are the cross product of the first two rows of R, since the rotation matrix is orthonormal and right-handed (determinant of R is 1). The second part estimates a single radial lens distortion parameter k_1, the principal distance c and the z-component t_z of the translation vector. Therefore, for every point correspondence another equation is set up:

$$\begin{bmatrix} Y_i & -d_y y_i' \end{bmatrix} \begin{bmatrix} c \\ t_z \end{bmatrix} = w_i d_y y_i' \tag{2.28}$$

where:

$$y_i = r_{21}X_i + r_{22}y_i + r_{23} \cdot 0 + t_y$$

$$w_i = r_{31}X_i + r_{32}y_i + r_{33} \cdot 0$$

d_y is the distance of sensor elements in y-direction. The linear system of equations can be resolved for f and t_z using a sufficient number of calibration points to obtain an over-determined system. The solution is refined by non-linear optimization (e.g. Levenberg-Marquardt-Algorithm [Ran04]) by using the following formula:

$$\begin{aligned} d_x x_i' + d_x x_i' k_1 r^2 &= c\frac{r_1X_i + r_2Y_i + r_3Z_i + t_x}{r_7X_i + r_8Y_i + r_9Z_i + t_z} \\ d_y y_i' + d_y y_i' k_1 r^2 &= c\frac{r_4X_i + r_5Y_i + r_6Z_i + t_y}{r_7X_i + r_8Y_i + r_9Z_i + t_z} \end{aligned} \tag{2.29}$$

where r is the distance of an image point to the image center and d_x the distance between sensor elements (pixels) in x-direction.

2.3.2 Direct Linear Transform and Zhang's Camera Calibration

Zhang's camera calibration method [Zha00] [GM04, pp. 5 – 43] basically uses a two dimensional calibration target (e.g. chessboard pattern). However, in [Zha04] the calibration is also shown with one dimensional target. Since specific camera orientations are necessary for the use of two and one dimensional targets, which is not possible within the practical part of this thesis (camera with two degrees of freedom), these targets and methods are not further described and can be revisited in [Zha04] and [Zha00]. Zhang's method is based on the so-called direct linear transform (DLT), which has the goal to estimate the elements of the projection matrix using known 3D coordinates. Similar to Tsai's calibration method, in a first step the targets' calibration markers are detected and assigned to their 3D world coordinate to set up 2D-3D point correspondences. From these correspondences, the projection matrix P can be estimated. Therefore, the direct linear transform [HZ03, p. 88] $\boldsymbol{x}_i = P\boldsymbol{X}_i$ with the homogeneous vectors $\boldsymbol{x}_i = (u_i, v_i, w_i)^T$ and $\boldsymbol{X}_i = (X_i, Y_i, Z_i, W_i)$ is established. The homogeneous vectors $\boldsymbol{x}_i$ and $P\boldsymbol{X}_i$ have the same direction but a varying magnitude. Thus, it's cross product equals to zero:

$$\boldsymbol{x}_i \times P\boldsymbol{X}_i = 0 \tag{2.30}$$

The complete form of equation 2.30 with

$$P\boldsymbol{X}_i = \begin{bmatrix} p^{1T}\boldsymbol{X}_i \\ p^{2T}\boldsymbol{X}_i \\ p^{3T}\boldsymbol{X}_i \end{bmatrix} \tag{2.31}$$

where p^{jT} denotes the j-th row of P (transposed four element vector), yields to:

$$\begin{bmatrix} y_i p^{3T}\boldsymbol{X_i} - w_i p^{2T}\boldsymbol{X}_i \\ w_i p^{1T}\boldsymbol{X_i} - x_i p^{3T}\boldsymbol{X}_i \\ x_i p^{2T}\boldsymbol{X_i} - y_i p^{1T}\boldsymbol{X}_i \end{bmatrix} = 0 \tag{2.32}$$

Rewriting equation 2.32 and separating the row vectors p^j results in:

$$\begin{bmatrix} \mathbf{0}^T & -w_i\boldsymbol{X}_i^T & y_i\boldsymbol{X}_i^T \\ -w_i\boldsymbol{X}_i^T & \mathbf{0}^T & -x_i\boldsymbol{X}_i^T \\ -y_i\boldsymbol{X}_i^T & x_i\boldsymbol{X}_i^T & \mathbf{0}^T \end{bmatrix} \begin{bmatrix} p^1 \\ p^2 \\ p^3 \end{bmatrix} = 0 \tag{2.33}$$

where $\mathbf{0}$ is an four element zero vector. For every point correspondence $\boldsymbol{x_i} \leftrightarrow \boldsymbol{X_i}$ the following equation is established, since only two of the three equations in the system 2.33 are linearly independent:

$$\underbrace{\begin{bmatrix} 0 & 0 & 0 & 0 & -X_i & -Y_i & -Z_i & -1 & y_iX_i & y_iY_i & y_iZ_i & y_i \\ X_i & Y_i & Z_i & 1 & 0 & 0 & 0 & 0 & -x_iX_i & -x_iY_i & -x_iZ_i & -x_i \end{bmatrix}}_{G_i} p = \mathbf{0} \tag{2.34}$$

where $p = [p_{11}, p_{12}, \ldots, p_{34}]^T$ are the elements of the camera's projection matrix. Using N points, a $2N \times 12$ matrix $G = [G_1^T, G_2^T, \ldots, G_N^T]^T$ is built. The System $G \cdot p = \mathbf{0}$ can be solved by singular value decomposition. This solution is an initial guess for a subsequent optimization using the Levenberg-Marquardt algorithm. Therefore, the non-linear lens distortion should be considered as well during the optimization to obtain a proper mapping between the detected and projected calibration markers. One can get to the projected calibration markers by transforming the known homogeneous 3D coordinates of the calibration markers by the projection matrix P and the distortion function. The optimization basically is a minimization of the error between detected and projected marker coordinates:

$$P = \min_P \sum_i d\left(\boldsymbol{x}_i, P\boldsymbol{X}_i\right)^2 \tag{2.35}$$

While Tsai's calibration method directly optimizes the camera parameters, the optimization in the DLT method delivers the compound projection matrix which needs to be decomposed into the rotation matrix R, the calibration matrix K and the translation vector $\vec{t}$, if those single parameters are required. Thus, the projection matrix P is considered as:

$$P = [M| - Mt] = K[R| - Rt] \tag{2.36}$$

K and R (with $M = KR$) are obtained by an RQ-decomposition of M, which results in an upper-triangular and orthogonal matrix. The camera center $\boldsymbol{C}$ and thus the translation vector $\vec{t}$ can be calculated from the last column of P ($\boldsymbol{p}_4$) by multiplying it with the inverse of $-M$ [HZ03, p. 163]:

$$t = -M^{-1} \cdot \boldsymbol{p}_4 \tag{2.37}$$

In [LRKB13] Luhmann et al. remove the lens distortion by correcting the image coordinates until the calibration markers match the projected coordinates using the perspective camera model and known 3D coordinates. Another approach is given by Thormälen et al. [TBW03], where markers that establish a line are considered to match a line in the image too. Due to lens distortion these lines are warped. The distortion function and its parameters are optimized until the warpage is removed.

2.4 Image Processing and Analysis

2.4.1 General

The following sections explain important basics in image processing and analysis that are necessary to understand the methodology and development of the work in this thesis. Besides the introduction to region properties, contour moments, integral images and the texture features of Haralick, which are indispensable methods for the pre-processing steps of the developed inspection system, essential methods for feature point extraction, description and matching are presented within the main part of this section. These algorithms are essential for the subsequent 3D reconstruction.

2.4.2 Region Properties and Contour Moments

A segmented image region can be described using region properties and moments. Generally, the moments, also called contour moments m and central moments μ, are defined by the following formulas:

$$m_{p,q} = \sum_{i=1}^{k} \sum_{j=1}^{l} I(x_i, y_j)\, x_i^p y_j^q \tag{2.38}$$

$$\mu_{p,q} = \sum_{i=1}^{k} \sum_{j=1}^{l} I(x_i, y_j)\, (x_i - \overline{x})^p (y_j - \overline{y})^q \tag{2.39}$$

In equation 2.38 a moment m includes the sum of the pixels within the considered binary contour image I, where the intensity value (1 or 0) of the pixel at (x, y) is multiplied by a factor, determined by x^p, y^q. Thus, in the binary image I the m_{00} moment represents the number of segmented pixels. $(\overline{x}, \overline{y})$ in equation 2.39 is the centroid of the corresponding region [KB16] [Jä12].

$$\overline{x} = \frac{m_{10}}{m_{00}} \; ; \overline{y} = \frac{m_{01}}{m_{00}} \tag{2.40}$$

2.4.3 Integral Images

Integral images are a simple concept to accelerate image processing steps where the sum of a rectangular region is needed. It can be used for example for linear filtering (e.g. mean filter) and plays an important role in feature detectors where box filters (e.g. approximations of second order Gauss kernels) are used. An integral image is setup up using equation 2.41:

$$I_{Integral}\left(x, y\right) = \sum_{i=0}^{x} \sum_{j=0}^{y} I\left(i, j\right) \tag{2.41}$$

Thus, a pixel in the integral image contains the sum of the underlying rectangular region between the current pixel and the origin. In conclusion, the value of a current pixel p_c can be obtained from the $(x-1, y)$, $(x, y-1)$ or $(x-1, y-1)$ pixel p_p by adding the remaining pixels of the current row, column or both. For the $(x-1, y)$ pixel one obtains

$$p_c = p_p + \sum_{j=0}^{y_{p_c}} I(i, j) \mid i = x_{pc} = x_{pp} \tag{2.42}$$

The sum over an arbitrary region in the original image can then be obtained by only three operations. This is shown in figure 2.14 and expressed by [KB16]:

$$\Sigma = \Gamma - \mathrm{B} - \Delta + \mathrm{A} \mid \mathrm{A} = \mathrm{B} \bigcup \Delta \tag{2.43}$$

2.4.4 Haralick Features

Object classification is a common task in computer vision and image processing. When a segmented region is examined, its properties and features are used to determine a class the region or object belongs to. In 2D image processing the texture of an object can be used to perform this classification. However, the term "texture" is not uniquely defined, since its recognition is subjective and can have a strong variation. Mathematical measures are needed to reconstruct and measure a surface [Toe12]. An approach to describe a texture is given by the texture parameters of Haralick, which are based on the gray level co-occurrence matrix (GLCM) C_δ. This is a representation of gray scale gradients along a specific geometric consideration. Depending on a displacement vector δ (see figure 2.15) every pixel in the image is examined. The displacement vector defines the relation of two pixels to be considered. The pixel combination is added to the co-occurrence matrix. An

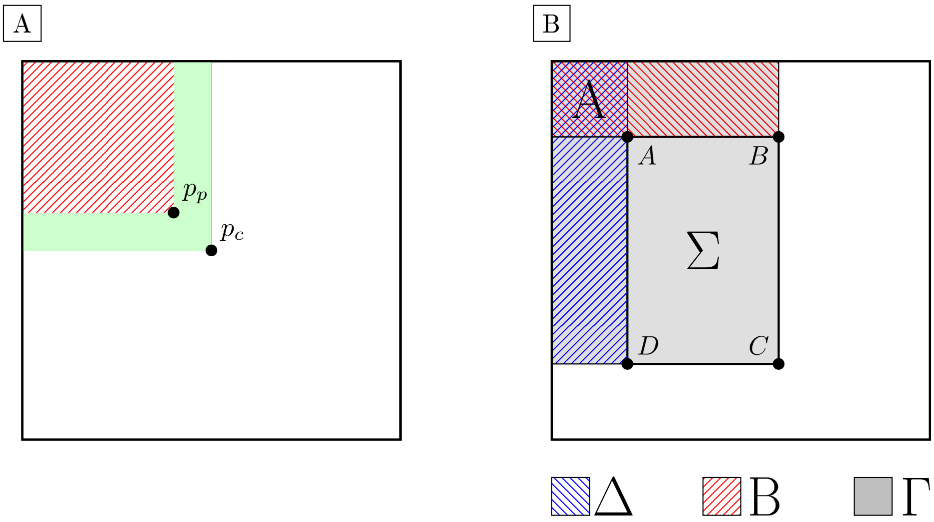


Figure 2.14 Integral image; A: the value of a pixel p_p of the integral image contains the sum of the pixels in the red hatched area of the original image. Another point p_c can be calculated by taking the previous point p_p and adding the current row and column pixel values; B: the sum of a region Σ can be calculated using integral images by subtracting the values at D and B from C and adding A once, since A=B$\bigcup$Δ [cf. KB16].

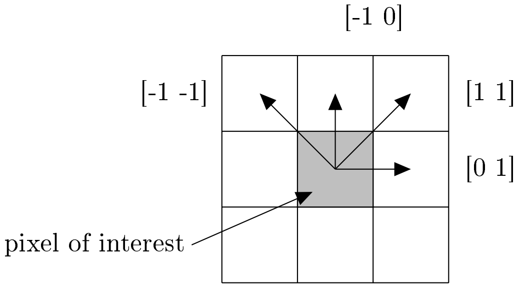
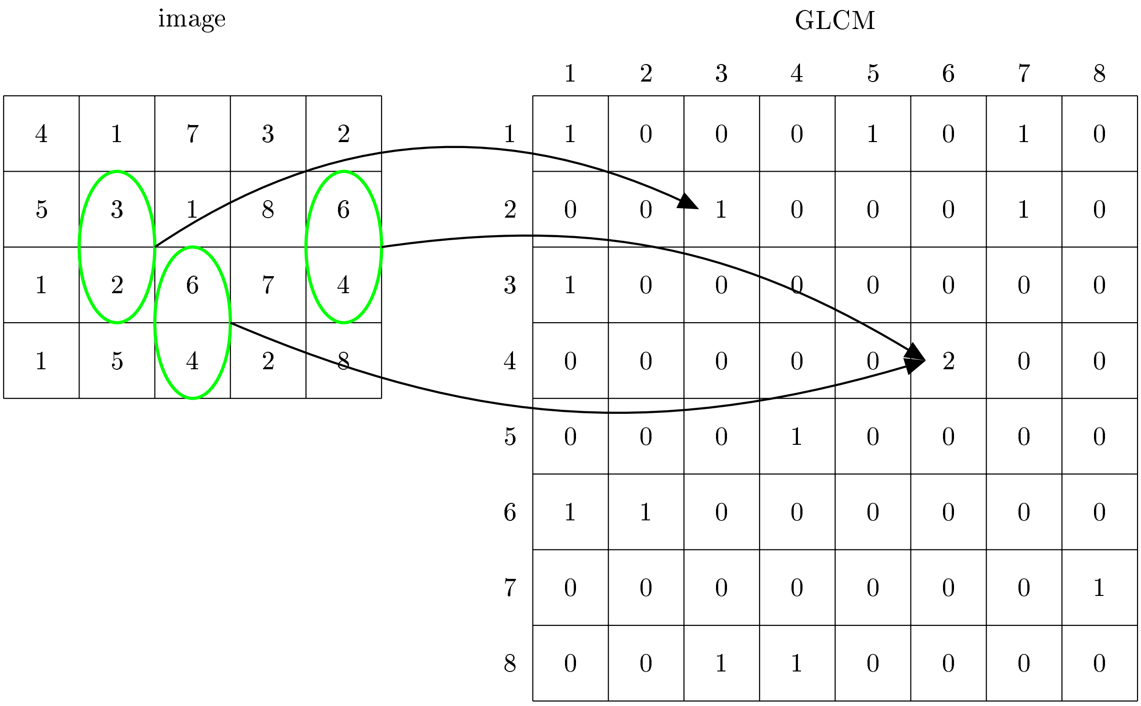


Figure 2.15 Displacement Vector: Starting from the pixel of interest the displacement vector defines the corresponding pixel. Here four examples are given [cf. Han09].

example of an 20 pixel image with values from 1 to 8 and the corresponding co-occurrence matrix for the displacement vector $(-1, 0)$ can be seen in figure 2.16. This specific vector defines the relation "above a pixel". Since the combination "6 above 4" occurs two times in the image, this cell gets the value two. Since image regions with different size can contain the same structure, the co-occurrence matrices of regions containing the structure can have a high variation. Therefore, the co-occurrence matrix is normalized by equation 2.44.

$$P_\delta = \frac{C_\delta}{\sum\limits_{k=0}^{g-1} \sum\limits_{l=0}^{k} C_\delta(k, l)} \tag{2.44}$$

where P_δ is the normalized co-occurrence matrix, g is the maximum gray value (e.g. 255) and k and l are the row and column indices of the co-occurrence matrix. Based on the co-occurrence matrix the Haralick texture parameters can be computed [Han09]. All 14 parameters *Contrast, Correlation,*



4	1	7	3	2
5	3	1	8	6
1	2	6	7	4
1	5	4	2	8

	1	2	3	4	5	6	7	8
1	1	0	0	0	1	0	1	0
2	0	0	1	0	0	0	1	0
3	1	0	0	0	0	0	0	0
4	0	0	0	0	0	2	0	0
5	0	0	0	1	0	0	0	0
6	1	1	0	0	0	0	0	0
7	0	0	0	0	0	0	0	1
8	0	0	1	1	0	0	0	0

Figure 2.16 Setup of the co-occurrence matrix: left: pixel values of an image; right co-occurrence matrix of the displacement vector (-1,0) [cf. Han09]

Information Measures of Correlation 1 and 2, Maximal Correlation Coefficient, Sum Average, Entropy, Sum Entropy, Difference Entropy, Variance, Sum Variance, Difference Variance, Second Angular Moment, Inverse Difference Moment are defined in appendix A.2.

2.4.5 Feature Detection

When 3D reconstructions from images are performed, point correspondences between distinctive image points across the used images need to be set up. Feature detectors are applied to the image to detect these points, which are called features, feature points or keypoints. The aim of a feature detector and descriptor is to find distinctive points in the image and to describe them in a way, to localize them in another image of the same scene, acquired with a variation in camera orientation and location. The description of a feature point is done by the feature descriptor and is used to match the features between two images. The most algorithms consist of a modular setup where detection and description can be used independent from each other and thus, can be combined across different algorithms. In the following sections the examined feature detectors, descriptors and matching methods are explained.

2.4.5.1 Scale Invariant Feature Transform (SIFT) The SIFT detector, proposed by Lowe [Low99] [Low04], is one of the best known and most used feature detectors, since it provides very stable results in almost every environment, although it is computationally expensive [KB16, p.539]. It yields feature points, that are invariant to scaling, translation, rotation, restricted illumination changes and affine or 3D projections. Basically, the algorithm consists of four steps (two for the detector and two for the descriptor):

- detector
  - scale-space extrema detection
  - keypoint localization
- descriptor
  - orientation assignment
  - descriptor calculation

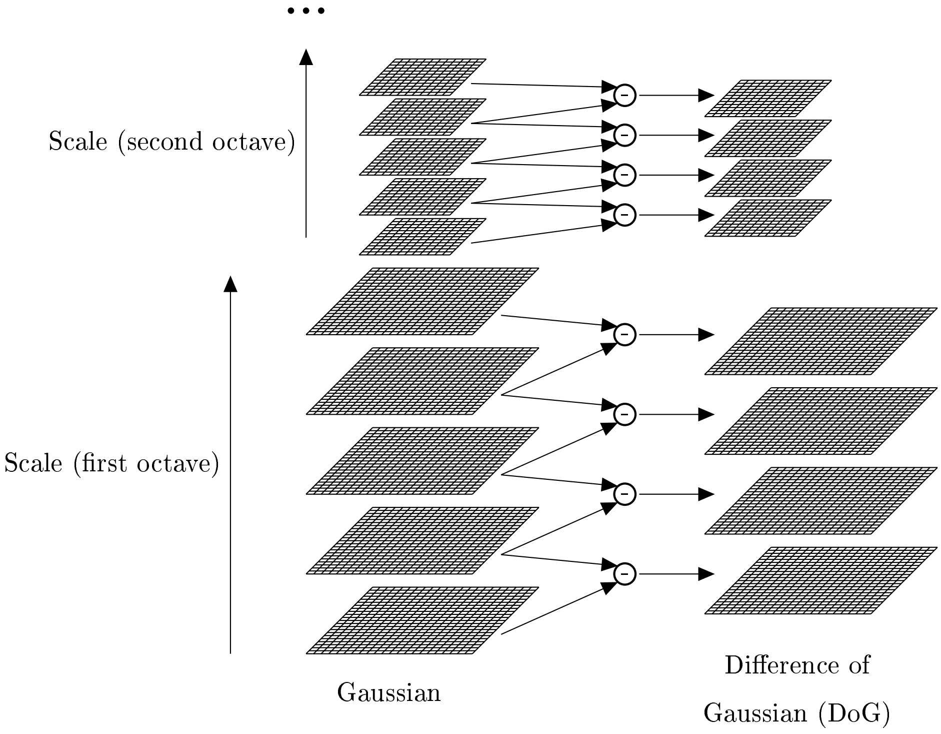


Figure 2.17 Scale and octave setup of SIFT: The input image is convolved with an increasing Gaussian kernel to obtain a stack of blurred images (left stack). Adjacent images are subtracted to set up the difference of Gaussian stack (right stack) [cf. Low99].

Scale-space Extrema Detection In the first part, the scale-space extrema detection, a staged filtering is used to detect stable keypoint in image space. As shown in Figure 2.17, the algorithm sets up so-called octaves and scales. A scale L is built from the input image, convoluted with a Gaussian kernel. Within an octave more than one scales are set up by increasing the size of the Gaussian kernel with a multiplicative factor k.

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y) \tag{2.45}$$

with

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-\left(x^2+y^2\right)/2\sigma^2} \tag{2.46}$$

Between each scale the difference of Gaussians (DoG) is computed by subtracting the two successive scales, resulting in a stack of DoG:

$$D(x, y, \sigma) = (G(x, y, k\sigma) * I(x, y) - G(x, y, \sigma) * I(x, y) \quad = L(x, y, k\sigma) - L(x, y, \sigma) \tag{2.47}$$

The number of octaves and scales depends on the image size and quality. Lowe [Low04] suggest five scales and four octaves for the most application cases. However, the user has to revise these settings and may adjust the numbers to fit to the actual task. In each layer of the DoG every pixel is compared to it's 26-neighborhood. That means, at first it is compared to its regular 8-neighborhood. If the pixel is a minimum or maximum of that region it's compared to the nine adjacent pixels in the previous DoG-level and if it's still an extremum, it's finally compared to the nine adjacent pixels of the next DoG-level. The pixel is marked as a potential feature candidate if it remains a maximum or minimum within the 26 pixel neighborhood.

Feature Point Localization In the first SIFT version [Low99] the obtained pixel coordinates where used as the feature point. However, the position was refined in the final version by fitting a paraboloid to the 27 pixel volume around the candidate position. From this fit the interpolated location of the maximum can be determined. Afterwards, the offset between the candidates coordinate and the interpolated location is obtained, as well as the local curvature at this pixel position by the Hessian matrix, to remove points at edges[Low04]. The last two steps for setting up the descriptor will be explained in section 2.4.6.1.

2.4.5.2 Speeded Up Robust Features (SURF) The SURF algorithm is based on SIFT. The main idea was to revise SIFT and to accelerate processing parts of it. The resulting detector is not just faster but also more robust against changes in rotation and illumination. The method takes advantage of integral images (section 2.4.3) and is a Hessian matrix $\mathcal{H}$ based detector. A feature is detected, if the determinant $\det(\mathcal{H})$ of this matrix becomes maximum.

$$\mathcal{H} = \begin{bmatrix} L_{xx}(\boldsymbol{x}, \sigma) & L_{xy}(\boldsymbol{x}, \sigma) \\ L_{xy}(\boldsymbol{x}, \sigma) & L_{yy}(\boldsymbol{x}, \sigma) \end{bmatrix} \tag{2.48}$$

In equation 2.48 L are the convolutions of the Gaussian partial second order derivative in x-, y- and xy- direction. E.g. for the x-direction:

$$L_{xx} = \frac{\partial^2}{\partial x^2} g(\sigma) \tag{2.49}$$

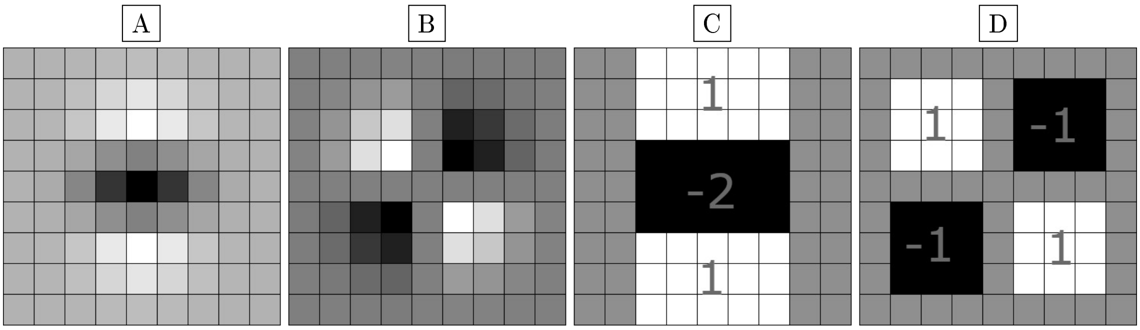


Figure 2.18 Approximation of second order filter kernels: A: discretized kernel L_{xx}, B: discretized kernel L_{xy}, C: approximated box filter D_{xx} of L_{xx}, D: approximated box filter D_{xy} of L_{xy} [cf. BTVG06].

The Hessian matrix needs to be approximated with box filters as depicted in figure 2.18 to provide the usage of integral images for the purpose of a simple and fast computation, . Let D_{xx}, D_{xy}, D_{yy} be the approximations, then the determinant of the Hessian matrix for $\sigma = 1.2$ and a 9×9 Kernel is given by:

$$\begin{aligned} \det\left(\mathcal{H}_{approx}\right) &= D_{xx}D_y - \left(wD_{xy}\right)^2 \\ w &= \frac{|L_{xy}\left(1.2\right)|_F |D_{yy}\left(9\right)|_F}{|L_{yy}\left(1.2\right)|_F |D_{xy}\left(9\right)|_F} = 0.912 \simeq 0.9, \end{aligned} \tag{2.50}$$

w is a weight which is needed to balance this expression, since the approximated kernels are different from the real kernels and $|x|_F$ is the Frobenius norm. Usually w should change over the scales, however, Bay et al. [BETG08] discovered no significant change in the results and thus, it remains constant. Similar to SIFT, also a scale space is built. However, instead of a subsequent Gaussian filtering and sub-sampling, the integral image is used with different filter sizes to obtain the scale space. The initial scale ($\sigma = 1.2$) is set up by the 9×9 box filter from figure 2.18. The filter size is increased to calculate multiple scales for an octave. The new filter size depends on l_0, which is the length of the positive or negative lobe in the previous filter. At the 9×9 kernel l_0 is three. The kernel needs to increase by one pixel at each side to remain at an odd kernel size and to ensure to have the reference pixel centered. As shown in figure 2.19 the filter increases by 6 pixel to preserve the kernel size. The first and the last scale layer can't contain any maximum, since these layers are used only for comparison reasons. Thus, the first maximum can be detected at a scale of 1.6. For another octave the increase of the filter size is doubled (e.g. from 6 to 12). The second octave starts at a filter size of 15, the third at 27 and so on. At each maximum an non-maximum suppression within the surrounding $3 \times 3 \times 3$ region is performed. Afterwards a similar interpolation as already explained in section 2.4.5.1 is done to calculate a more precise location of the feature point [BTVG06] [BETG08].

2.4.5.3 Features from Accelerated Segment Test (FAST) The FAST detector was developed by Rosten and Drummond [RD06] and is based on pixel comparisons. The basic idea is, that a pixel represents a feature point if its value differs (is brighter or darker) from the values

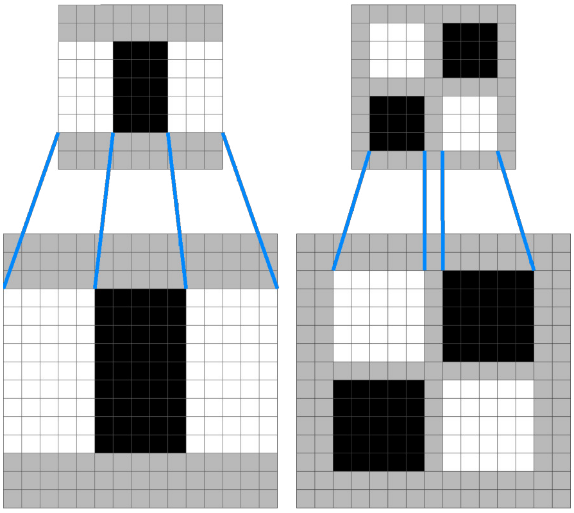


Figure 2.19 SURF: Increase of the box filter size: An increase of 6 pixel is needed to preserve the kernel size from the left D_{xx} to the D_{xy} kernel [cf. BTVG06].

of a set of pixels in a surrounding circle with a user defined radius as shown in figure 2.20. A feature point is detected if $N/2+1$ adjacent pixels have a brighter or darker value compared to the candidate pixel, where N is the number of pixels on the circle. The pixels 1, 5, 9 and 13 are evaluated first to speed up the detection. If at least three of them are brighter plus a user defined threshold t or all of them are darker minus this threshold, the other pixels are examined. Since the neighbor pixels of the candidate may also satisfy this condition a non-maximum-suppression is introduced, where a score (see equation 2.51) is calculated for every candidate pixel. Candidate pixels that are neighbors to a pixel with a higher score a removed.

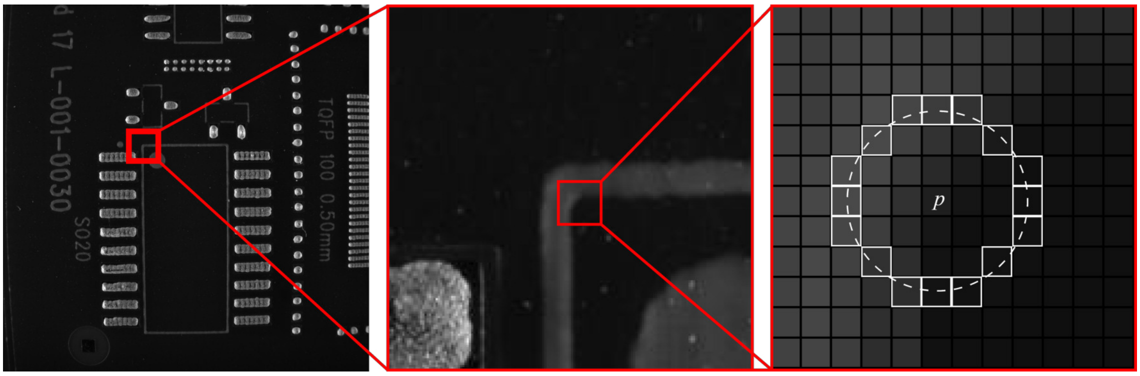


Figure 2.20 FAST detector: left: original image, middle: enlarged region, right: enlarged region with pixel segment test of 9-16 configuration. The corresponding pixel p is a feature point if 9 of 16 pixels (white) in the dashed circle have a higher or lower value compared to p.

$$score = max\left(\sum_{x\in\{brighter\}} |I_x - (I_p + t)|, \sum_{x\in\{darker\}} |I_x - (I_p - t)|\right) \tag{2.51}$$

The score is computed by taking the maximum of the sum of absolute differences between brighter pixels and the sum of absolute differences of darker pixels. In some cases $N/2+1$ adjacent brighter or darker pixels lead to features that are not necessarily good features. Therefore, the detector can be configured to e.g. 9-16 configuration, which means 9 (adjacent) out of 16 pixels need to satisfy the condition. This configuration is much more stable and thus can be found more often in practical applications. The FAST algorithm does not provide a descriptor.

2.4.5.4 Binary Rotation Invariant Scalable Keypoints (BRISK) This detector is based on the FAST algorithm, but instead of calculating feature candidates and scores only on the original image a scale space pyramid is set up with octaves c_i and intra-octaves d_i with n layers $(i = 1...n)$, where n typically is set to 4. The different octaves and intra-octaves are obtained by half-sampling the previous (intra-) octave, with exception of the first intra-octave d_0 which is calculated by a downsampling with the factor of 1.5. If s is the scale, then $s(c_i) = 2^i$ and $s(d_i) = 2^i \cdot 1.5$. The FAST algorithm is performed on each octave and intra-octave using the 9-16 configuration. The same non-maximum suppression as explained in the previous (FAST) section, is performed. Thus, a pixel must have the maximum FAST-score within its 8-neighborhood to be a feature point. In addition the scores in the layers below and above within the adjacent 3×3 pixel pattern must be smaller then the score of the selected point. In each of these three layers a 2D quadratic fit is done to refine the position of the maximum within the mentioned 3×3 patterns. Since the maximum of the three layers does not necessarily lay exact on a scale layer a 1D parabola fit is performed to find the real maximum position between the scales. Finally, the image coordinates are interpolated from the adjacent imaged scales of the maximum found by the 1D fit. The sequence of this processing is visualized in figure 2.21 [LCS11].

2.4.5.5 Oriented FAST and Rotated BRIEF (ORB) ORB is basically a further development of the FAST detector, where an orientation of the FAST feature is introduced. After the computation of FAST features with a radius of 9 pixels , the detected points are weighted by a Harris corner measure, which is based on the co-variance matrix at a feature point. From this matrix the eigenvalues λ_1 and λ_2 are calculated and the measure of the cornerness R is given by [HS88]:

$$R = \lambda_1 \lambda_2 - k\left(\lambda_1 + \lambda_2\right)^2 \tag{2.52}$$

The keypoints are sorted by this measurement, thus, if a number N of keypoints is required by the user, the strongest ones, in terms of the cornerness are returned.
For each point the orientation is calculated, based on the intensity centroid C:

$$C = \left(\frac{m_{10}}{m_{00}}, \frac{m_{01}}{m_{00}}\right) \tag{2.53}$$

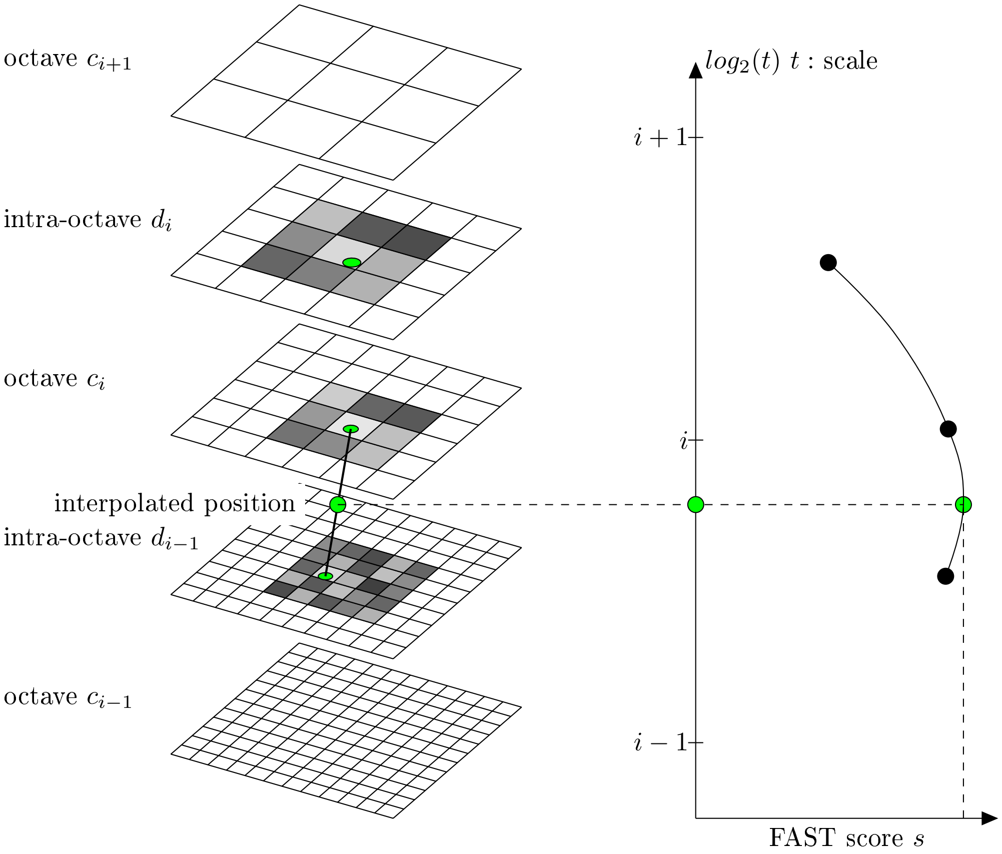


Figure 2.21 Principle of the BRISK detector: left - downsampling of the image to obtain octaves and intra-octaves, small elliptical green dots represent a feature position along different (intra-) octaves the large green dot is the final interpolated inter scale space feature position; right - scale space quadratic fit to the FAST score s [cf. LCS11].

.

Then a vector $\vec{OC}$ from the corner center O to the centroid C is computed and the orientation angle θ is given by:

$$\theta = \arctan 2\,(m_{01}, m_{10}) \tag{2.54}$$

[RRKB11]

2.4.5.6 Maximally Stable Extremal Regions (MSER) MSER is a feature detector whose function is comparable to the watershed transform. The image is thresholded with all possible thresholds (e.g. 8 bit gray scale : 255 thresholds). The first image is completely white and the last image is black. In between these minimum and maximum values small regions (local minima) arise when the threshold increases. At some point these regions will merge, where the smaller region is eliminated and added to the larger one. Therefore, the pixel intensities are sorted and inserted into the image by their value. In each step a list of connected components is extracted. The obtained data structure holds the area of the connected components in dependency of the intensity. Those intensity values, where the rate of change in the area function becomes minimal, define a threshold which produces a maximally stable extremal region. Compared to the watershed, the MSER output is different. Instead of finding thresholds where regions get connected, MSER estimates thresholds where the watershed basins remain. Depending on the image content some regions may consist of multiple, nested regions [MCUP04].

2.4.6 Feature Descriptors

Feature descriptors examine the neighborhood of a detected feature point to create a distinctive description of it. These descriptors are essential for the subsequent feature matching. Feature detectors that do not contain a descriptor part must be used in combination with an descriptor extractor of another detector.

2.4.6.1 Scale Invariant Feature Transform (SIFT)

orientation assignment At each extracted feature point the gradient orientation $\theta(x, y)$ and magnitude $m(x, y)$ is calculated in the smoothed image L, where the keypoint was located. Therefore, the directional derivatives of the surrounding point in a typically 16×16 pixel array around the feature points are calculated:

$$
\begin{aligned}
m(x,y) &= \sqrt{\left(L\left(x+1,y\right) - L\left(x-1,y\right)\right)^2 + \left(L(\left(x,y+1\right) - L\left(x,y-1\right)\right)^2} \\
\theta(x,y) &= \tan^{-}1 \frac{\left(L\left(x,y+1\right) - L\left(x,y-1\right)\right)}{\left(L\left(x+1,y\right) - L\left(x-1,y\right)\right)}
\end{aligned}
\tag{2.55}
$$

From the orientations a 36 bin orientation histogram is created. A points orientation is weighted by a Gaussian kernel ($\sigma = 1.5 \cdot \sigma_{scale}$), covering the 16×16 pixel area and by its magnitude. The dominant orientation is used for the orientation of the keypoint. If there are other orientations with a peak of at least 80% of the dominant orientation, further keypoints are created with the corresponding orientation. Thus, each keypoint is defined by its location, scale and orientation.

Descriptor calculation Finally the local feature descriptors are set up. For orientation invariance the coordinates of the sample region and the corresponding orientations are rotated relative to the main orientation found previously. The 16×16 pixel region is divided into 4-by-4 blocks, where a histogram of oriented gradients (HoG) is built. This results in a block of 4×4 HoG, each containing 8 orientations. This arrangement is converted to a vector representation containing $4 \times 4 \times 8 = 128$ elements. This vector is also called feature vector [Low04].

2.4.6.2 Speeded Up Robust Features (SURF) Similar to SIFT, the descriptor extractor of SURF computes an orientation component. However, instead of calculating gradient information between various sample points, Haar wavelets in combination with the integral image are used. The wavelets with the size of $4s$ (s = scale) are applied in a circular region of $6s$ around the detected point. The responses are weighted with a feature point centered Gaussian ($\sigma = 2s$) and are plotted in a new coordinate system, where the horizontal responses are placed along the abscissa and the vertical responses along the ordinate. A $pi/3$ sliding window sums up the two parts of the responses (x- and y- direction), which results in a final vector. Afterwards, the vector of maximum length is extracted as the orientation of the key point. The final descriptor is built from a rectangular region with a side-length of $20s$ around the feature point. The region is rotated by the detected orientation and is divided into 4×4 subregions, where in each sub-region 5×5 sample points are distributed uniformly. At each point the Haar wavelet responses d_x and d_y are calculated and weighted with another Gaussian (centered at feature point, $sigma = 3.3s$). The responses in each subregion, as well as the corresponding absolute values are summed up resulting in a 4D vector for each region.

$$v = \left(\sum d_x, \sum d_y, |\sum d_x|, |\sum d_y|\right). \tag{2.56}$$

Thus, the whole 4×4 region has a descriptor length of 64 [BTVG06] [BETG08].

2.4.6.3 Binary Rotation Invariant Scalable Keypoints (BRISK) The descriptor part of the BRISK algorithm also considers the orientation of a keypoint. A specific neighborhood sampling is done to obtain the final descriptor pattern and the orientation. As shown in figure 2.22 circles with circumference C_i are spanned around the keypoint. On each circle K_i sample points $\boldsymbol{p}_i$ are distributed equally as the centers of further circles with the diameter of C_i/K_i. Prior to sampling of these points a Gaussian filtering is performed with $\sigma_i = C_i/2K_i$. Afterwards the local image gradient $\boldsymbol{g}$ is calculated as follows:

$$\boldsymbol{g}\left(\boldsymbol{p}_i, \boldsymbol{p}_j\right) = \left(\boldsymbol{p}_j - \boldsymbol{p}_i\right) \cdot \frac{I\left(\boldsymbol{p}_j, \sigma_j\right) - I\left(\boldsymbol{p}_i, \sigma_i\right)}{\|\boldsymbol{p}_j - \boldsymbol{p}_i\|^2} \tag{2.57}$$

where $\boldsymbol{p}_i, \boldsymbol{p}_j$ is a pair of sampling points, whereas $I(\boldsymbol{p}_i, \sigma_i)$ and $I(\boldsymbol{p}_i, \sigma_i)$ are the smoothed intensity values of these points. For the calculation of orientation and descriptor pattern, the method introduces short and long distance point pairs $\mathcal{S}$ and $\mathcal{L}$ which are determined by the distance threshold $\delta_{max} = 9.75s$ and $\delta_{min} = 13.67s$ (with s = scale of the point):

$$\begin{aligned} \mathcal{S} &= \{\left(\boldsymbol{p}_i, \boldsymbol{p}_j\right) | \|\boldsymbol{p}_j - \boldsymbol{p}_i\| < \delta_{max}\} \\ \mathcal{L} &= \{\left(\boldsymbol{p}_i, \boldsymbol{p}_j\right) | \|\boldsymbol{p}_j - \boldsymbol{p}_i\| > \delta_{min}\} \end{aligned} \tag{2.58}$$

From the long distance pairs the pattern direction is calculated:

$$\boldsymbol{g} = \begin{pmatrix} g_x \\ g_y \end{pmatrix} = \frac{1}{\mathcal{L}} \cdot \sum_{(p_i, p_j) \in \mathcal{L}} \boldsymbol{g}\left(\boldsymbol{p}_i, \boldsymbol{p}_j\right). \tag{2.59}$$

Consequently, the short distance point pairs are used to setup the descriptor pattern. Previously, the sampling pattern is rotated by the obtained direction ($\alpha = \arctan2\left(g_y, g_x\right)$) around the key-point. The descriptor itself is a bit-vector, where each bit is set to one or zero:

$$b = \begin{cases} 1, & I\left(\boldsymbol{p}_j^\alpha, \sigma_j\right) > I\left(\boldsymbol{p}_i^\alpha, \sigma_i\right) \\ 0, & otherwise \end{cases} \forall \left(\boldsymbol{p}_i^\alpha, \boldsymbol{p}_j^\alpha\right) \in \mathcal{S} \tag{2.60}$$

Under this configuration the feature vector has a length of 512 bit and can be modified by the change of the distance thresholds [LCS11].

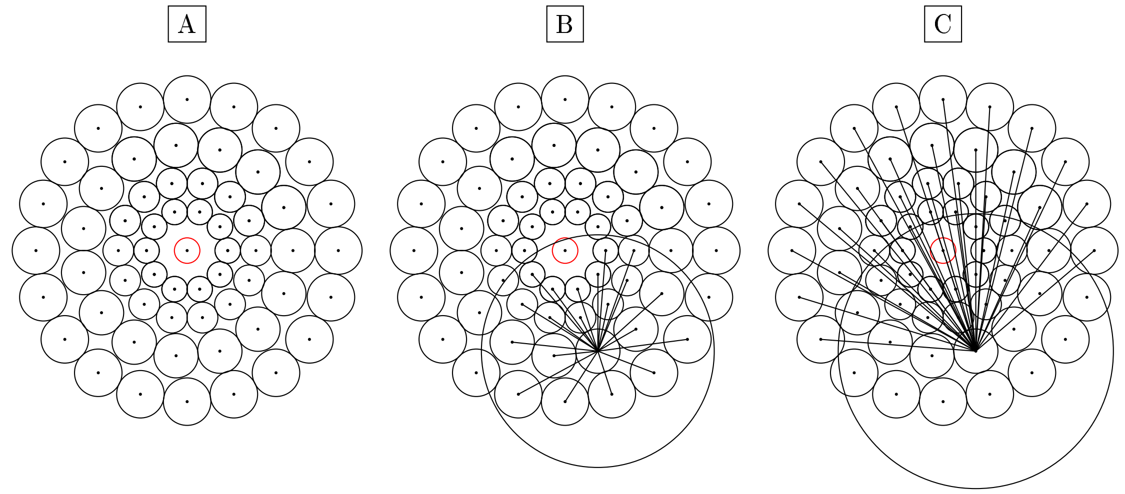


Figure 2.22 Principle of the BRISK descriptor: A: general test point layout; B: connection of point pairs with distance $\delta < \delta_{max}$ for the calculation of the descriptor pattern; C: connection of point pairs with distance $\delta > \delta_{min}$ for the calculation of the orientation [KB16].

2.4.6.4 Oriented FAST and Rotated BRIEF (ORB)

Computing a descriptor within the ORB algorithm is based on principle of the BRIEF descriptor extractor. BRIEF [CLSF10] is short for *Binary Robust Independent Elementary Features*, which uses simple pixel comparisons as a binary test τ (see eq. 2.61) within a region around a detected feature point. The points can be distributed randomly by setting the first points within a Gaussian distribution around the feature point and the second point within a Gaussian distribution around the first.

$$\tau\left(\boldsymbol{p};\boldsymbol{x},\boldsymbol{y}\right) := \begin{cases} 1 & : \boldsymbol{p}\left(\boldsymbol{x}\right) < \boldsymbol{p}\left(\boldsymbol{y}\right) \\ 0 & : \boldsymbol{p}\left(\boldsymbol{x}\right) > \boldsymbol{p}\left(\boldsymbol{y}\right) \end{cases} \tag{2.61}$$

$\boldsymbol{p}$ is the intensity of a point $\boldsymbol{x}$ or $\boldsymbol{y}$ respectively. The test results are concatenated to the feature vector, usually with a length of 256, whereas other lengths are also possible. However, ORB considers the orientation of a feature point and a matrix $\boldsymbol{S}$ which holds the point pairs:

$$\boldsymbol{S} = \begin{pmatrix} \boldsymbol{x}_1 & \ldots & \boldsymbol{x}_n \\ \boldsymbol{y}_1 & \ldots & \boldsymbol{y}_n \end{pmatrix} \tag{2.62}$$

This matrix is rotated by a rotation matrix constructed from the orientation angle θ and is called a steered version $\boldsymbol{S}_\theta$ of $\boldsymbol{S}$. Finally, the algorithm searches for proper binary tests in a way, that the mean is close to 0.5 and the tests being uncorrelated. Therefore, all possible tests are performed, ordered by the distance of the mean value from 0.5 and stored in a vector T. Then, a greedy search is performed. The first element of T is transferred to the result vector R. Afterwards the algorithm iterates through T and compares the current element against all elements in R. The element is put into the result vector if the correlation is above a predefined threshold. This iteration is performed until R holds 256 elements. If this size can't be reached the threshold of the correlation is increased.

2.4.7 Descriptor Matching

2.4.7.1 General After the detection and description of feature points the next goal is to determine point-to-point correspondences. That means for a feature point in image A, a corresponding point in image B needs to be found. The matching methods in the next sections are used to find these correspondences. A correct correspondence between two points is also called a match. Thus, the determination of point correspondences is called feature matching.

2.4.7.2 Brute Force The *brute force* matching is the simplest method for finding feature matches. It is simply a comparison of every descriptor in feature vector A with every descriptor in feature vector B. For the decision, if a descriptor pair is a match, various distance metrics (see table 2.3) are used to compute the vector distance of the descriptors, to measure its similarity.

Table 2.3 Brute force matching: metrics

Metric	Function
ℓ_2 - Norm	$d\left(\vec{a},\vec{b}\right) = \left\lvert \sum_i (a_i - b_i)^2 \right\rvert^{1/2}$
ℓ_2 - squared Norm	$d\left(\vec{a},\vec{b}\right) = \sum_i (a_i - b_i)^2$
ℓ_1 - Norm	$d\left(\vec{a},\vec{b}\right) = \sum_i \lvert(a_i - b_i)\rvert$
Hamming - Norm	$d\left(\vec{a},\vec{b}\right) = \sum_i c_i \mid c_i = \begin{cases} 1; a_i = b_i \\ 0 \text{ otherwise} \end{cases}$
Hamming 2 - Norm	$d\left(\vec{a},\vec{b}\right) = \sum_i c_i \mid c_i = \begin{cases} 1; a_i = b_i \wedge a_{i+1} = b_{i+1} \\ 0; \text{ otherwise} \end{cases} \quad \mid i \mod 2 = 0$

2.4.7.3 FLANN *FLANN* is short for Fast Library for Approximate Nearest Neighbor and is a collection of various algorithms for high-dimensional nearest neighbor searches. These methods construct an index of the feature vectors to efficiently find matches. If the construction of the index is done linear, the method corresponds to the brute force matching. The basic nearest neighbor search becomes computationally expansive in large and high-dimensional data sets. Therefore, the sets are prepared in a way to perform an efficient search, which allows for an increase of speed in orders of magnitude. The three main techniques to do so are partitioning trees, hashing techniques and neighboring graph techniques. The specific algorithms will not be explained in detail. Therefore, it will be referred to the work of Muja and Lowe in [ML12], [ML14] and [ML09].

The explained matching methods work well in a natural environment where the descriptors of the feature points have a high variation. However, in industrial manufacturing, especially in PCB jet printing, the descriptors become very similar, since the surface structure of a solder joint is almost across of different solder joints. That's why the feature matching results in a high number of false matches. These lead to significant errors in the reconstruction and must be eliminated prior to the triangulation. This will be explained in section 4.2.4.

2.4.8 Point Cloud Reconstruction

There are many methods to reconstruct an object from 2D image feature points, like the bundle adjustment [TMHF00] in structure from motion (SfM) [SF16]. However, these methods are inappropriate for the practical solution presented later. For the reconstruction of detected and matched feature points a linear stereo triangulation is performed (see figure 2.23 for the ideal stereo triangulation). The cameras in figure 2.23 are oriented to form the photogrammetric normal case,

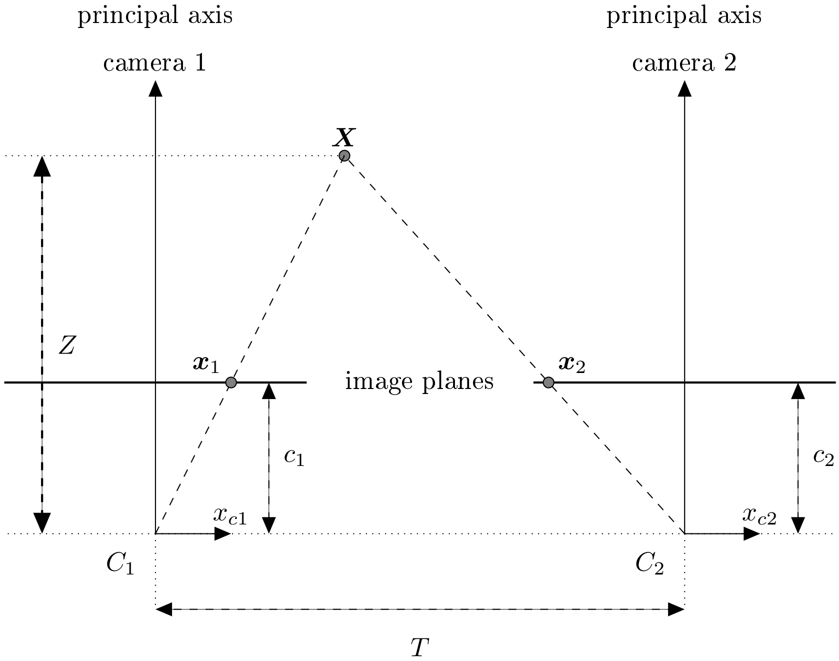


Figure 2.23 Normal case triangulation: depending on the position of corresponding points $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ and via the camera model the position of the world point $\boldsymbol{X}$ can be estimated via triangulation (dashed lines). C_1 and C_2: camera centers; c_1 and c_2: principal distances; T: translational distance between camera centers; Z: working distance [cf. LRKB13].

where the principal axes of the cameras (or the shifted camera) are parallel and the image planes are within the same plane. The linear triangulation can be described as the solution of a linear system of equations. A point $\boldsymbol{X}$ satisfies the condition of triangulation if it projects to the image points $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ that belong to the estimated point correspondence:

$$\begin{aligned} \boldsymbol{x}_1 &= P_1 \boldsymbol{X} \\ \boldsymbol{x}_2 &= P_2 \boldsymbol{X} \end{aligned} \tag{2.63}$$

with P_1 and P_2 being the projection matrices of the two camera positions. From these equations a form $A\boldsymbol{X} = 0$ can be set up. Each image point delivers three equations, whereas only two of them are linear independent. For one image this results in:

$$\begin{aligned} x_1 \left(\boldsymbol{p}_1^{3T} \boldsymbol{X}\right) - \left(\boldsymbol{p}_1^{1T} \boldsymbol{X}\right) &= 0 \\ y_1 \left(\boldsymbol{p}_1^{3T} \boldsymbol{X}\right) - \left(\boldsymbol{p}_1^{2T} \boldsymbol{X}\right) &= 0 \\ x_1 \left(\boldsymbol{p}_1^{2T} \boldsymbol{X}\right) - y_1 \left(\boldsymbol{p}_1^{1T} \boldsymbol{X}\right) &= 0 \end{aligned} \tag{2.64}$$

the rows of P are denoted by $\boldsymbol{p}^{iT}$. Rearranging this system for both images results in:

$$A = \begin{pmatrix} x_1 \boldsymbol{p}_1^{3T} & - & \boldsymbol{p}_1^{1T} \\ y_1 \boldsymbol{p}_1^{3T} & - & \boldsymbol{p}_1^{2T} \\ x_2 \boldsymbol{p}_2^{3T} & - & \boldsymbol{p}_2^{1T} \\ y_2 \boldsymbol{p}_2^{3T} & - & \boldsymbol{p}_2^{2T} \end{pmatrix} \tag{2.65}$$

This system can be solved for example by using the singular value decomposition as already described in section (2.2). Due to localization and matching errors the correspondence $\boldsymbol{x}_1 \leftrightarrow \boldsymbol{x}_2$ may not be precise. This in turn leads to deviations in the reconstructed point coordinates. A geometric error minimization is used to correct these errors. Let the initial reconstructed point be donated as $\hat{\boldsymbol{X}}$ that projects to the point $\hat{\boldsymbol{x}_1}$ and $\hat{\boldsymbol{x}_1}$. Then a projection error between the detected feature points $\boldsymbol{x}_1$ and projected point $\hat{\boldsymbol{x}_1}$ can be calculated. This error is minimized to find the best point that satisfies the system of equations 2.63 [KB16, p.704-708] [HZ03, p.312].

$$\boldsymbol{X} = \min_{\hat{\boldsymbol{X}}} \left(d\left(\boldsymbol{x}_1, \hat{\boldsymbol{x}_1}\right)^2 + d\left(\boldsymbol{x}_2, \hat{\boldsymbol{x}_2}\right)^2 \right) \mid \hat{\boldsymbol{x}_1} = P_1 \hat{\boldsymbol{X}} \text{ and } \hat{\boldsymbol{x}_2} = P_2 \hat{\boldsymbol{X}} \tag{2.66}$$

2.5 Photogrammetry

Photogrammetry is a method to estimate a 3D model (point clouds) from 2D images of a scene or object. Thus, it is a method to measure shape and position of objects indirectly without contact. That's why photogrammetry belongs to the measurements of remote sensing and has a lot of advantages compared to other inspections like laser or moire based methods. For example the images of the object contain more information than point positions or lengths like regional relations and color information of the object. The acquisition can be done with low effort. At least two images of object of interest are needed to reconstruct and measure a surface [MMB04] [GL84]. For the reconstruction the imaging process, more specific the geometric image projection needs to be modeled. This is done by camera calibration which yields a camera model, that describes the projection of a 3D world point onto the image plane [LRKB13]. Depending on the measured object and the corresponding imaging setup photogrammetry is typically divided into the application fields of *aerial*, *close range* [Mat08] and *ultra-close range*, which is newly defined in this thesis. While *aerial* defines the application of cameras in aircrafts or unmanned aerial vehicles (UAV), *close range* usually considers a non-topographic photogrammetry [Gra16]. However, the application fields are not clearly separated. Thus, *close range* applications can also consider small UAV's. The dimensions of *close range* are not well defined in the literature. Some define any application with a working distance below 300 meters to be close range [Mat08], others consider distances in a lower range of 1-4 meters [Sco76], [Dua71]. An application where one or more cameras acquire images of an object or surface without significant alteration of the working distance and orientation (the camera main axes are perpendicular to the examination surface and the parallel to each other) is considered to be the photogrammetric *normal case* [LRKB13], [Gra16]. 3D reconstruction and virtual reality, mobile mapping, industrial and engineering measurement, architectural and archaeological as well as forensic investigation are examples where photogrammetry has become an indispensable analysis tool. [MMB04, pp. 1015]. Figure 2.24 shows the typical photogrammetric process.

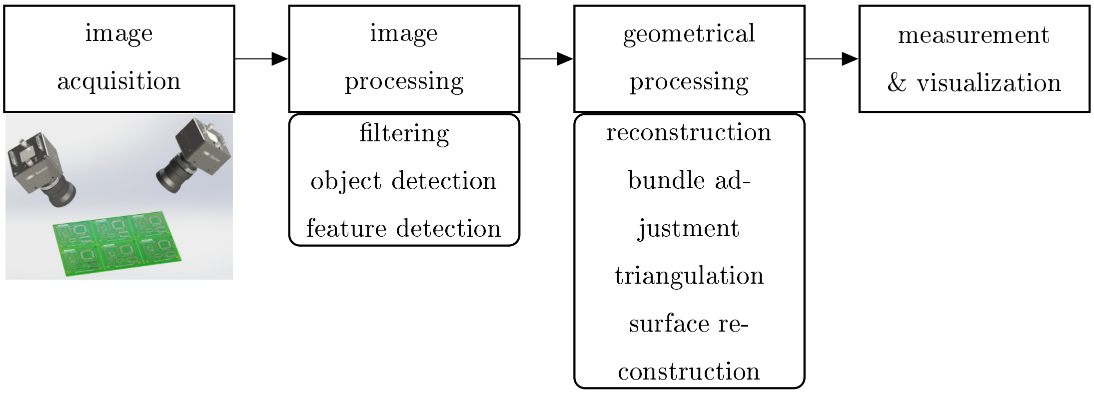


Figure 2.24 Photogrammetric process

3 Ultra-Close Range Normal Case Camera Calibration

3.1 General

Starting with this chapter, the methodology of this thesis is presented. In section 2.5, the application fields and dimensions of photogrammetry were mentioned. In this work, the novel term *ultra-close range normal case*, with a working distance in the range of 5 to 100 mm and following the definition of *normal case*, is introduced to clearly separate this special application from state of the art photogrammetry.

Standard image processing and calibration toolboxes (like openCV) use Zhang's calibration method and chessboard or circular pattern calibration targets [Zha00]. However, these targets and methods require multiple views of the calibration target under different camera orientations [Zha04], [Zha99], which is not applicable within ultra-close range normal case applications. Herein, the camera calibration process becomes a challenging task since the camera has restricted degrees of freedom, due to its fixation (i.e. the camera cannot rotate around any axis and cannot perform any translation along the z-axis), a short working distance and a large magnification using spacer rings or extension tubes in between the lens and the sensor, which also results in a small depth of field. In addition, the user needs to recalibrate the camera as soon as the imaging setup changes (usually a variation in working distance and magnification in this application), without the extraction of the camera. In this work, threaded high-precision lenses enable the adjustment of the lens-sensor distance without the detachment of the lens from the camera. Those restrictions and a postulated ease of use result in a non-applicability of the standard calibration methods and patterns.

In this thesis, a camera calibration approach and a corresponding calibration target in a millimeter range without any rotation of the camera or the calibration target is proposed. This procedure is called *ultra-close range normal case camera calibration*. It can be adopted to other 3D calibration targets and ensures a precise camera calibration. A high precision of the calibration is essential for subsequent measurements or reconstructions.

Zhou et al. [ZN99] presented a calibration method based on the model of an optical microscope. In both the considered imaging setup of Zhou et al. and the demonstrator system of this work (see section 5), the sensor plane is nearly parallel to the examination plane and a large magnification is obtained by extension tubes. Zhou's method was adopted to the laboratory setup and the lens model was extended with up to fkeypointsradial distortion coefficients and two decentering coefficients to examine the applicability of the proposed method for the setup.

The distortion correction methods of Luhmann et al. [LRKB13] and Thormälen et al. [TBW03] (see section 2.3) require the assignment of the markers to their real 3D coordinates or to a line. This thesis proposes a geometry based distortion correction where no such assignments are necessary, since no projections or shape assumptions (markers on a line) are necessary and the geometric relation between the markers is optimized. For the subsequent calibration, single and multi-image methods were implemented and evaluated within different imaging geometries obtained from different combinations of working distances and spacer rings. The evaluation was done by examining the reprojection error ϵ_{repro_s} for single images or the compound reprojection error ϵ_{repro_m} when using multiple images as well as the reconstruction error ϵ_{recon}. For that purpose, a new calibration target (figure 3.1 A) was designed and manufactured.

The operating distances (200 – 3000 mm) of the used lenses are not appropriate for ultra-close range photogrammetry. Therefore, different sensor-lens distances (spacers) are examined to ensure suitable operating distances. The targets were imaged with the lenses and spacers to examine the applicability of the setups for the subsequent reconstruction. The image processing and calibration methods have been evaluated using 700 images, acquired by a gray scale camera.

3.2 Calibration Targets

A 3D calibration target was specifically developed for this application since no satisfying commercially available target could be found for this task. For 2D calibrations a commercially available target was used. A single image calibration for a planar (2D) target and single and multi-image calibration for the 3D target where the calibration markers are distributed along x-, y-, and z-direction were performed.

The 2D target (Figure 3.1 B) consists of circular chromium markers on glass within three areas, where the markers have a diameter of 1.0, 0.5 and 0.25 mm with a spacing of 2.0, 1.0 and 0.5 mm. The markers are distributed on a single plane.

Figure 3.1 A shows the 3D calibration target. For 3D camera calibrations 3D object information is needed to obtain sufficient camera models. Therefore, calibration markers (white dots) are distributed stepwise on three height levels. For the purpose of origin and orientation determination, the target also includes three central special markers (square, triangle, and cross). The calibration markers have a spacing of 2.5 mm on each level, while the levels have a spacing of 0.5 mm. The target was manufactured using cnc-milling of an aluminum block. Afterwards it was anodized black and laser labeled to reduce reflections as much as possible. The dimensions of the target where chosen to cover the whole field of view of the camera for larger working distances and to yield 3D information within the depth of field of the camera which does not exceed 1 mm in the ultra-close range (use of large spacers). The 3D target was examined using a structured light measurement to affirm its accuracy and ensure that no geometric distortions are present. The system has a resolution of 22.4 μm in x- and y- direction and an accuracy of 2 μm in z- direction. The result of the measurement is shown in figure 3.2. Randomly selected profile lines at every target level where examined, which show no deviations in its height exceeding 30 μm (see figure 3.3). Detailed engineering drawings of the 3D target can be found in the appendix on pages a to c in figure A.1 ,A.2 and A.3.

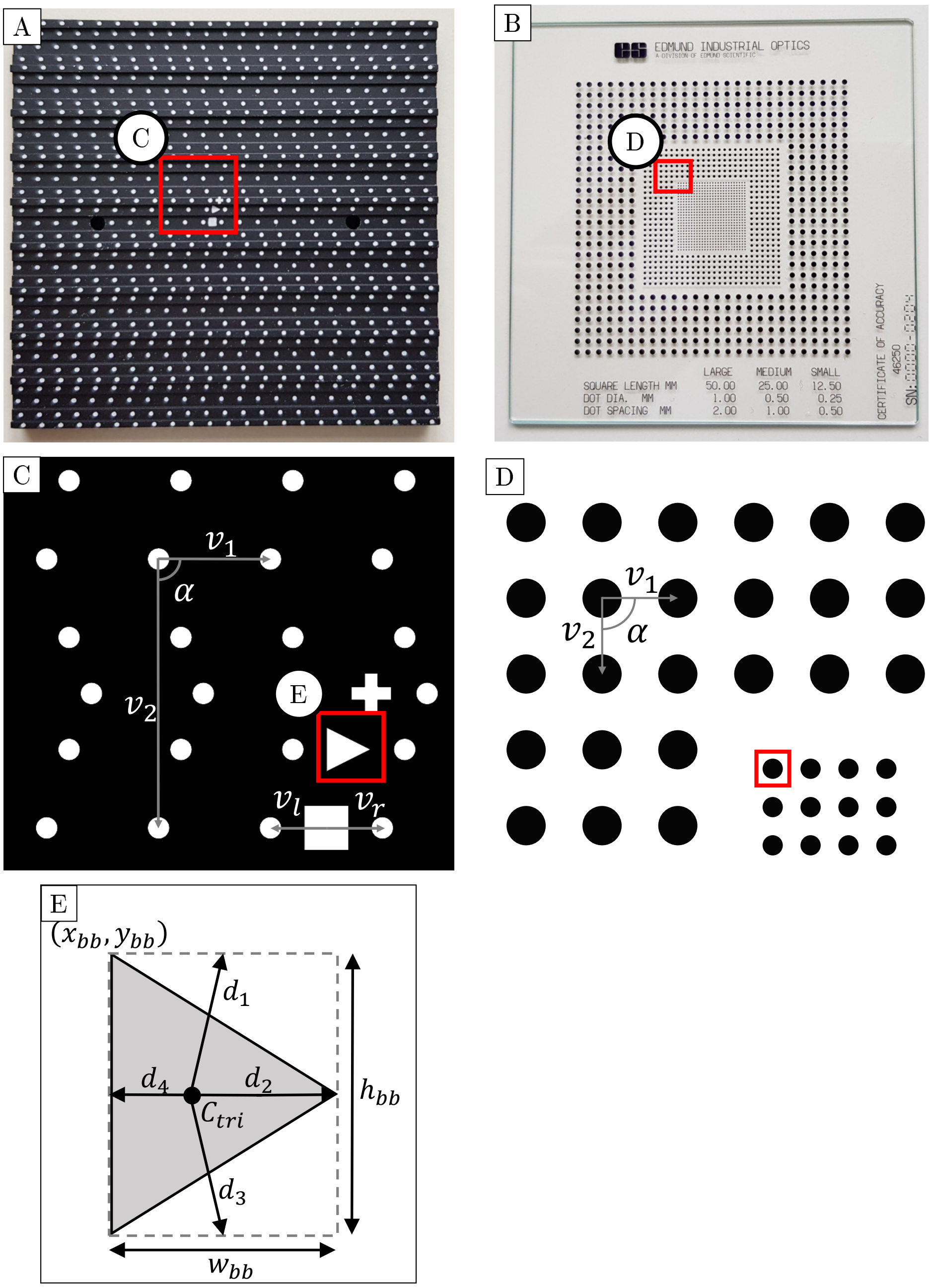


Figure 3.1 3D (left) and 2D target (right): A: photograph of the 3D calibration target (top view); B: photograph of the 2D calibration target (top view) - red squares indicate enlarged view areas in C and D; C: v_1 and v_2 are vectors within the 3D target to the two nearest neighbors of a marker that build a 90° angle α_{ij}; v_l and v_r are vectors to the two nearest neighbors of the square marker (the red square indicates the enlarged view area in E); D: v_1 and v_2 are the same vectors as described for the 3D target, the upper left marker of the inner most set of markers (red rectangle) is used as origin. E: geometric considerations of the triangle marker; dashed box: bounding box of triangle marker; (x_{bb}, y_{bb}): origin of the bounding box; $d1, d2, d3, d4$: distances from the centroid to the centers of bounding box edges; C_{tri} - centroid of the triangle; h_{bb} - bounding box height; w_{bb} - bounding box width.

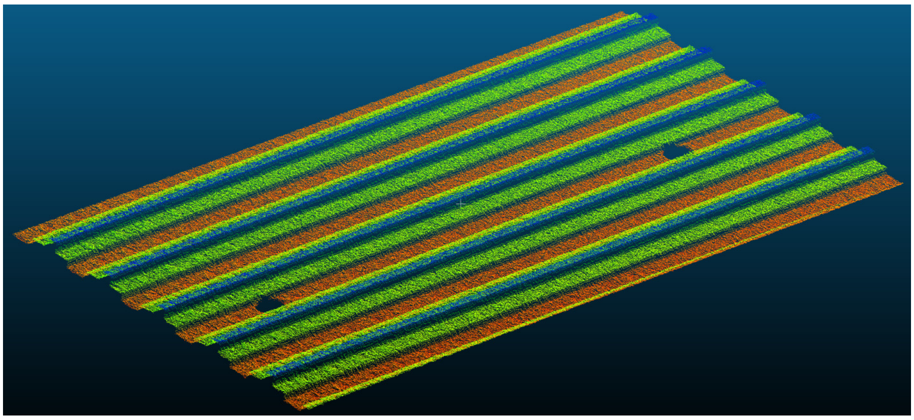


Figure 3.2 3D Scan of 3D Target - different colors express the height profile of the target.

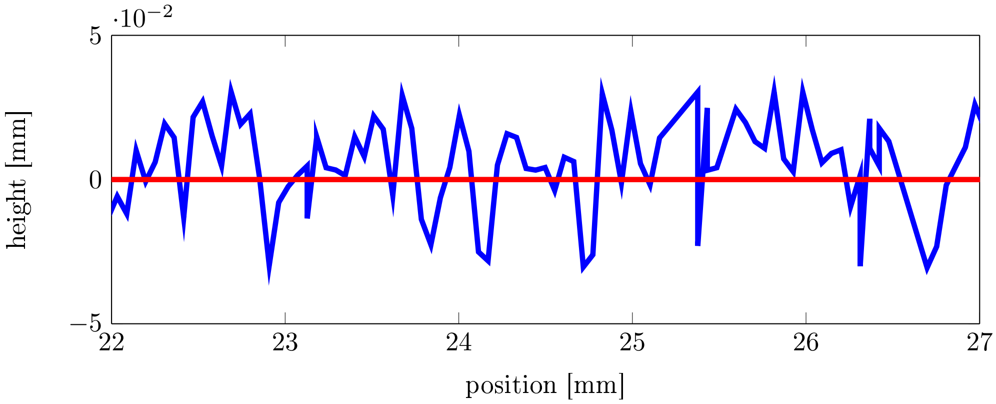


Figure 3.3 Exemplary Profile Line (blue) of the 3D Targets zero level (red line).

3.3 Target and Marker Detection

Before the camera calibration process itself can take place, various pre-processings are needed to obtain 3D-2D point correspondences, the correspondence between a 3D-world point $\boldsymbol{X}$ and the related 2D-image point $\boldsymbol{x}$, respectively.
For the purpose of automation within the jet printing inspection system the calibration target needs to be detected automatically. The detection is based on the geometric arrangement of the calibration markers. The developed detection approach works with arbitrary calibration targets where a known pattern is used. Since the calibration was done within different working distances and magnifications some images can contain artifacts, which are produced by dust, scratches or reflections. These artifacts may appear in certain imaging configurations, while others deliver suitable images of the calibration target. The detection consists of the steps indicated in figure 3.4 to handle the variety of image scales and artifacts.

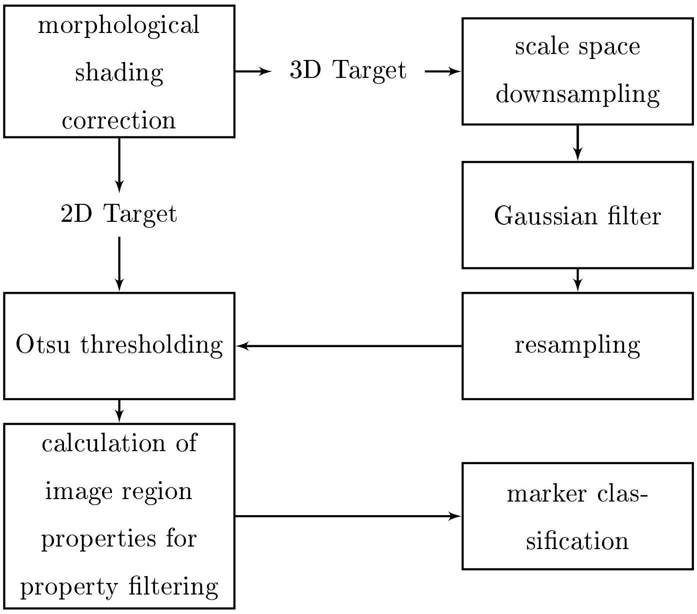


Figure 3.4 Flowchart of the image pre-processing.

Morphological Shading Correction The irregular artifacts need to be treated to obtain algorithms that are as much universal as possible. Figure 3.5 shows example artifacts at different working distances and magnifications. At first, a morphological opening based shading correction is used to remove significant gray scale gradients, which results in a homogeneous background with clearly visible white calibration markers. The morphology estimates the background, which is subtracted from the input image [KB16].

$$I_c = I - I \circ S = I - (I \ominus S) \oplus S \tag{3.1}$$

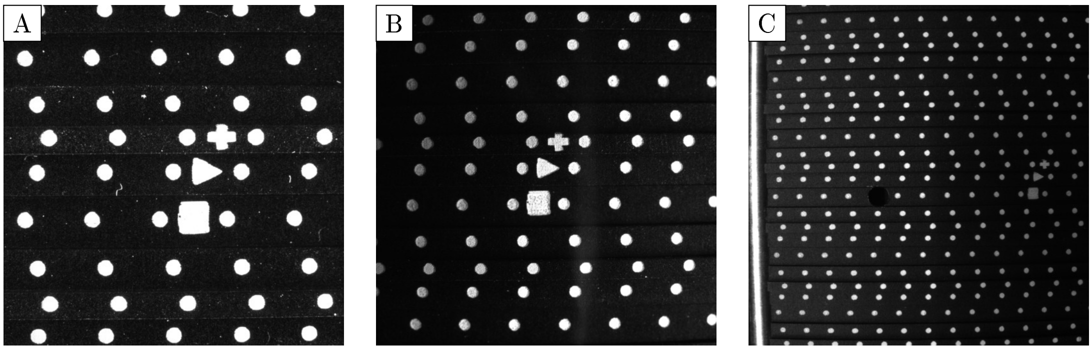


Figure 3.5 Examples of artifacts, A: short working distance, dust and scratches cause unwanted artifacts; B: medium working distance, artifacts by dust and scratches become less visible, shading gradient appears; C: large working distance, dust particles are not visible anymore, large shading gradient is present.

Scale Space Downsampling The scale space sampling and subsequent filtering deliver advantages compared to a single scale image in the detection of markers and elimination of artifacts in the image. In each step the image I_c with m columns and n rows is downsampled by the factor of two to obtain the lower scale image I_s:

$$I_s(x, y) = I_c(2x, 2y) \mid 0 \geq x > m;\ 0 \geq y > n \tag{3.2}$$

Gaussian Filtering On the downsampled image I_s a Gaussian filter [Lee83] is applied to remove remaining artifacts. The filter is defined by the two-dimensional Gaussian function in equation 3.3 [LRKB13]:

$$I_f(x, y) = \frac{1}{2\pi\sigma^2} \cdot e^{\left(-\frac{x^2+y^2}{2\sigma^2}\right)} \mid \sigma = 2 \tag{3.3}$$

Resampling The downsampled and filtered image has to be resampled to get correct marker coordinates. Therefore, a bilinear interpolation is used to retain the resolution of the original images.

Otsu Thresholding After the filtering of the downsampled image Otsu's segmentation method [Ots79] is used to divide the image into two classes (markers and background) and to create a binary image. Since most of the artifacts are eliminated the Otsu threshold is the simplest, fastest and most efficient way to extract the markers. The method calculates an optimal threshold t to divide the mentioned classes.

$$I_b = \begin{cases} 0, \text{ if } I_f < t \\ 1, \text{ if } I_f \geq t \end{cases} \tag{3.4}$$

Calculation of Image Properties For the subsequent classification of the extracted segments, the moments of the regions in combination with the underlying original image are calculated to obtain the following image properties:

area: The area of a region corresponds to the non-zero pixels and thus, to the m_{00} moment of the binary image.

weighted centroid: The weighted centroid c_w of a certain region is calculated using the m_{00}, m_{01} and the m_{10} moments. Using formula 2.38 results in:

$$c_{wx} = \frac{\sum_{i=1}^{N} I\left(x_i, y_i\right) x}{\sum_{i=1}^{N} I_b\left(x_i, y_i\right)} \tag{3.5}$$

for the x-coordinate of the weighted centroid and

$$c_{wy} = \frac{\sum_{i=1}^{N} I\left(x_i, y_i\right) y}{\sum_{i=1}^{N} I_b\left(x_i, y_i\right)} \tag{3.6}$$

for the y coordinate.

minor and major axis length: The minor and major axis lengths of a region are defined by the equivalent ellipse and associated with the eigenvalues λ_1 and λ_2 of the matrix A:

$$A = \begin{bmatrix} m20 & m11 \\ m11 & m02 \end{bmatrix} \tag{3.7}$$

The corresponding lengths r_a (major) and r_b (minor) are given through equations 3.8 and 3.9[BB15, p. 257]:

$$r_a = 2 \cdot \left(\frac{\lambda_1}{|\mathcal{R}|}\right)^{\frac{1}{2}} \tag{3.8}$$

$$r_b = 2 \cdot \left(\frac{\lambda_2}{|\mathcal{R}|}\right)^{\frac{1}{2}} \tag{3.9}$$

$\mathcal{R}$ is the number of pixels within a Region.

eccentricity: In the subsequent detection and classification of the calibration markers all circular segments are regions of interest. However, due to perspective distortions the markers may not appear as a perfect circle [Luh14]. Thus, the eccentricity $Ecc\left(\mathcal{R}\right)$, defined in equation 3.10, will not be exactly one, but gives a good criterion to exclude non-marker areas. The literature presents different definitions of the eccentricity, e.g. the ratio of the distance of the ellipses foci and the major axis length [MM15] or via the moments of a region [BB15, p. 256], which is used in this work:

$$Ecc\left(\mathcal{R}\right) = \frac{m_{20} + m_{02} + \sqrt{\left(m_{20} - m_{02}\right)^2 + 4 \cdot m_{11}^2}}{m_{20} + m_{02} - \sqrt{\left(m_{20} - m_{02}\right)^2 + 4 \cdot m_{11}^2}} \tag{3.10}$$

Marker Classification After the target is detected, the markers have to be classified (calibration marker and orientation marker for the 3D target; group of calibration markers - determined by diameter and distance to each other - for the 2D target) and the correspondence of each image coordinates $\boldsymbol{x}$ to its 3D world counterpart $\boldsymbol{X}$ needs to be set up for the subsequent calibration procedure. The previously mentioned image properties in combination with the geometric information of the calibration target is used to get the correct marker positions. In the first step all objects with a small area (less than 10 pixels) and with $Ecc > 2$ are removed. These areas usually represent most of the mentioned artifacts. Since markers, which are located at the edge of the image may not be completely visible, the calculated weighted centroids c_w do not correspond to the real center of mass, which will result in inappropriate point correspondences. Therefore, all regions with a bounding box touching the edge of the image are removed too. These steps remove the most artifacts. However, it may occur, that under certain circumstances, like a specific acquisition setup, some of the artifacts do remain, since they follow the upper properties for regions that should remain in the image. A-priori knowledge of the calibration target is used to prevent the usage of wrong marker positions. In the case of the 3D target this information is used to extract the additional markers (cross, rectangle, triangle). In an image without artifacts the rectangle should correspond to the largest area. A nearest neighbor search [FBF77] is used to find the two neighbor-markers $\mathcal{N}_1$ and $\mathcal{N}_2$ of the largest area and the vectors v_l and v_r between the centroids c and the sum vector v_s.

$$\begin{aligned} \vec{v}_l &= \begin{pmatrix} x_{v_l} \\ y_{v_l} \end{pmatrix} = c_{Rect} - c_{N1} \\ \vec{v}_r &= \begin{pmatrix} x_{v_r} \\ y_{v_r} \end{pmatrix} = c_{Rect} - c_{N2} \\ \vec{v}_s &= \vec{v}_l + \vec{v}_r \end{aligned} \tag{3.11}$$

Through these vectors and the region properties a decision can be made, if the selected area (maximum area) is the rectangle marker of the 3D target:

$$\mathcal{R}_{max} = \begin{cases} \mathcal{R}_{rect} & ;\text{if} \begin{array}{ll} r_a(\mathcal{R}_{max}) < 3 \cdot r_a(\mathcal{N}_1) & \text{and} \\ r_a(\mathcal{R}_{max}) < 3 \cdot r_a(\mathcal{N}_2) & \text{and} \\ |\vec{v}l| < 1.5 \cdot r_a(\mathcal{R}_{max}) & \text{and} \\ |\vec{v}_r| < 1.5 \cdot r_a(\mathcal{R}_{max}) & \text{and} \\ \frac{|\vec{v}_s|}{|\vec{v}_1|} < 0.1 & \text{and} \\ \frac{|\vec{v}_s|}{|\vec{v}_2|} < 0.1 & \end{array} \\ 0 & ;else \end{cases} \tag{3.12}$$

If one of the conditions is false, the region is neither a special nor a calibration marker, and is removed. The center of the rectangle defines the origin of the target and will be set to $(0,0,0)^T$. The primary orientation of the marker rows is given by $\vec{v}_1$ and $\vec{v}_2$. Depending on the target, a number of nearest neighbors (2D target: 5; 3D target: 30) is extracted for each marker. For every set pair of a current marker and k-nearest neighbors, the following parameters are calculated:

- vectors $|\vec{v}_i|$ and $|\vec{v}_{ij}| \mid i = [0;29]$ and $j = [i+1;30$ (3D); $i = [0;4]$ and $j = [i+1;5]$ (2D) between marker and neighbors similar to equation 3.11

- norm of the vectors: $|\vec{v}_i|$ and $|\vec{v}_{ij}|$
- angle α_{ij} between vectors $|\vec{v}_i|$ and $|\vec{v}_{ij}|$:

$$\alpha_{ij} = \frac{\arccos\left(\frac{\vec{v}_i \cdot \vec{v}_{ij}}{|\vec{v}_i| \cdot |\vec{v}_{ij}|}\right) \cdot 180}{\pi} \tag{3.13}$$

- relation of the minimum and maximum vector norm:

$$Rel_v = \frac{min\,(|\vec{v}_i|)}{max\,(|\vec{v}_i|)} \tag{3.14}$$

- angle *beta* between v_1 and the target direction v_r

$$\beta = \frac{\arccos\left(\frac{\vec{v}_i \cdot \vec{v}_r}{|\vec{v}_i| \cdot |\vec{v}_r|}\right) \cdot 180}{\pi}; \tag{3.15}$$

- relation of the minimum of norms and the norm of the direction vector v_r

$$Rel_o = \frac{min\,(|\vec{v}_i|, |\vec{v}_{ij}|)}{|\vec{v}_{l|r}|} \tag{3.16}$$

A proper marker has at least two pairs of neighbor vectors that form a 90° angle. Rel_v should be one for the 2D target and owing to the design, it should be 0.41 for the 3D target and 0.5 for the Rel_o. Owing to lens distortions, a tolerance of ± 10° is allowed for the angles α_{ij} and β, and ± 0.2 for the norm relation (equation 3.14). Otherwise, the region is removed. The characterizations of the markers for the two targets are as follows:

2D target:

$$\mathcal{R} = \begin{cases} \mathcal{R}_m & ;\text{if} \quad 80^\circ \leq \alpha_{ij} \leq 100^\circ \quad \text{and} \quad 0.8 \leq Rel_v \leq 1.2 \\ 0 & ;\text{else} \end{cases} \tag{3.17}$$

3D target:

$$\mathcal{R} = \begin{cases} \mathcal{R}_m & ;\text{if} \quad \begin{array}{lll} 80^\circ \leq \alpha_{ij} \leq 100^\circ & \text{and} \quad 0.21 \leq Rel_v \leq 0.61 & \text{and} \\ -10^\circ \leq \beta \leq 10^\circ & \text{and} \quad 0.3 \leq Rel_o \leq 0.7 & \end{array} \\ 0 & ;\text{else} \end{cases} \tag{3.18}$$

$\mathcal{R}$ is an extracted region (contiguous pixels) and $\mathcal{R}_m$ is a pixel region corresponding to a marker. For the 3D target, its orientation that can be determined by the triangle marker is important because the markers are assigned to their world coordinate depending on the orientation. The triangle is classified by the second largest region following the conditions in equation 3.12. Considering the centroid c_{tri} and the triangle's bounding box defined by its origin (x_{bb}, y_{bb}), width w_{bb}, and height h_{bb}, the minimum of all distances $(d_1 \ldots d_4)$ (see figure 3.1 E) between the centroid and the center of each bounding box edge defines the orientation.

At each marker, an ellipse fitting [FPF99] was performed to obtain a better estimation of the centroid, as the intensity-weighted centroid might be distorted owing to artifacts. When no markers or rectangles are detected, or the number of markers is insufficient for the calibration and distortion estimation, the next scale is processed. If this fails for all scales, the algorithm exits with the statement of poor image quality. After pre-processing, only the calibration markers are left in the image.

3.4 Calibration

3.4.1 Initial Distortion Correction

As shown in figure 3.1, the markers are arranged along lines. This arrangement was used to simplify the assignment between the markers' 2D and 3D coordinates. However, owing to lens distortions, the markers may not be imaged on the straight lines. Especially in images acquired with short focal lengths, this effect becomes problematic. The geometric relations of the calibration points are disturbed by lens distortion and can be optimized. Here up to fkeypointsradial distortion coefficients k_i are used. The coefficients were estimated by a nonlinear least-squares approach where the difference between the angles α_{ij} and 90° is minimized:

$$[k] = \min_{k_{1\dots4}} \left(\sum_i \sum_j |\alpha_{ij} - 90°|^2 \right) \tag{3.19}$$

After the optimization of the coefficients and the application of equation 2.20, the image contains the corrected coordinates, and the markers are aligned on straight lines.

3.4.2 Point Correspondences Setup

<u>*3D target:*</u> Initially, the lines where the markers were arranged on were extracted. The marker coordinates were rotated by the angle between the target's direction vector and the x-unit vector of the image coordinate system. The minimum y-distance between the lines on the target is 1.25 mm. Via the extracted rectangle, one can estimate the number of pixels per mm using the width or height of the rectangle. All markers within an y-interval of 1.25 mm were assigned to the same line. The z value is determined by the modulus of the rectangle's line index and four, owing to the distribution on three z-levels:

$$z = \begin{cases} 0 & \text{, if } l_i \mod 4 = 1 \\ 0.5 & \text{, if } l_i \mod 4 = \{0, 2\} \\ 1 & \text{, if } l_i \mod 4 = 3 \end{cases} \tag{3.20}$$

Owing to the target's layout, the coordinates of the markers in every extracted line can be calculated by geometric considerations.

<u>*2D target:*</u> The 2D markers are assigned similarly. Initially, the markers were separated by their size and each class was sorted by lines, as described previously. As no specific markers indicate the orientation or origin of the target, the upper left marker of the smaller dots (marked with a red

rectangle in figure 3.1 D) was used as the origin. Starting from this point and the known target dimensions, the extracted lines were used to assign each marker in the image to its 3D coordinates.

3.4.3 3D Single-Image Calibration

In general the projection matrix P is searched. Based on the DLT approach [KB16], the system in equation 2.5 with homogeneous coordinates $\boldsymbol{x}_i = [x_i, y_i, w_i]^T$ and $\boldsymbol{X} = [X_i, Y_i, Z_i, 1]^T$ is rearranged to:

$$\begin{pmatrix} 0^T & -w_i\boldsymbol{X}_i^T & y_i\boldsymbol{X}_i^T \\ w_i\boldsymbol{X}_i^T & 0^T & -x_i\boldsymbol{X}_i^T \\ -y_i\boldsymbol{X}_i^T & x_i\boldsymbol{X}_i^T & 0^T \end{pmatrix} \begin{pmatrix} p^1 \\ p^2 \\ p^3 \end{pmatrix} = 0 \Rightarrow A \cdot p = 0 \tag{3.21}$$

This system can be solved by the use of the singular value decomposition [KL80][HZ03].

$$A = U\Sigma V^T \tag{3.22}$$

In theory, this solution is correct. However, due to possible errors in the manufactured target and in the marker detection, the solution is not precise and is refined by minimizing the reprojection error ϵ_{repro_s} [HZ03] between the projected 3D coordinates and detected image coordinates using the nonlinear optimization method of Levenberg and Marquardt [Lev44] [Mar63].

$$\epsilon_{repro\_s} = \sum_{i=1}^{N} d\left(\boldsymbol{x}_i, P\boldsymbol{X}_i\right)^2 \tag{3.23}$$

For this purpose the image points are corrected using the parameters obtained in 3.4.1 using equation 2.22. Within the optimization all extrinsic, intrinsic and lens parameters are refined, resulting in a small reprojection error ϵ_{repro_s}. As an abbreviation for this method, SC for single image calibration, and SCCo for single image calibration, where the reconstruction result is compared with the corrected target of the section 3.4.5, will be used.

3.4.4 3D Multi-Image Calibration

During the examinations, an improper initialization using a single image was observed, thereby resulting in deviations in the camera model. In this specific application, owing to the small calibration target and the correlation of intrinsics and extrinsics, especially the principal distance and working distance, the reprojection error ϵ_{repro_s} does not increase significantly if the camera center, along with the principal distance, changes (e.g., the camera center comes closer to the target while the principal distance decreases). To handle this, a multi-image calibration is introduced, where overlapping images of the target are acquired, each with a normal case orientation. The initial model calculation and a first optimization were performed as described in section 3.4.3. However, to suppress the mentioned effect, another optimization, where the model for each camera of each single image must exhibit the same intrinsic parameters, is used. Thus, a second sum of the errors is introduced over the image acquisition position j.

$$\epsilon_{repro\_m} = \sum_{j=1}^{M}\sum_{i=1}^{N} d\left(\boldsymbol{x}_{ij}, P_j \boldsymbol{X}_{ij}\right)^2 \tag{3.24}$$

This method is abbreviated MC.

3.4.5 Iterative 3D Multi-Image Calibration

In the final adaption of the previous procedure, an iterative loop comparable to [ART09] is introduced. The optimized camera positions were used to reconstruct the target points using a linear triangulation without optimizing the camera parameters [TMHF00]. Owing to given uncertainties in the manufactured target, the modified calibration points are close to the real 3D coordinates of the calibration markers. These points are used to recalibrate the camera within all imaging positions. This loop was performed until a predefined reprojection error ϵ_{repro_m} or a fixed number of iterations was reached. The multi-image calibration using a corrected target is abbreviated MCCo.

3.4.6 2D Single-Image Calibration

For the calibration of the model of an optical microscope model the basics of the work of Zhou et al. [ZN99] and Pieters at al. [PJN10] with some adoptions and the mentioned 2D target is implemented. The determination of the intrinsic and extrinsic camera parameters is based on Tsai's calibration method [Tsa87]. While Tsai is using the center of the image as principal point, a more accurate solution with the geometric optimization proposed in 3.4.1 is obtained. The method requires an a-priori knowledge of the focal length, which is usually given by the manufacturers data. The method is abbreviated SC2D.

3.5 Model Evaluation

The different combinations of spacers and lenses lead to varying image scales and depth of field (DOF) as it is shown in table 3.1. In addition to the reprojection errors ϵ_{repro_s} and ϵ_{repro_m}, the mean reconstruction error ϵ_{recon} is used to evaluate the obtained models via a 3D reconstruction of the 2D calibration target of all setups from table 3.1. The obtained models are used to reconstruct the 2D target (calibration markers) and to calculate the total 3D reconstruction error ϵ_{recon} , as well as the error along the x-y plane and along the z direction.

$$\epsilon_{recon} = \frac{1}{N}\sum_{i=1}^{N} d\left(X_i, X_{recon}\right)^2 \tag{3.25}$$

The reprojection errors ϵ_{repro_s} and ϵ_{repro_m} were obtained by a synthetic projection of the known marker positions using the obtained camera models and the measured image coordinates using equations 3.23 and 3.24.

Table 3.1 Depth of field (DOF) and images scales for the examined combinations of lenses and spacer sizes

lens	spacer	DOF (μm)	image scale
10 mm	1.4 mm	235,078	30.303
	1.9 mm	58,770	15.1515
	2.4 mm	17,485	8.2645
	2.9 mm	8,264	5.6818
	3.4 mm	5,289	4.5455
	3.9 mm	3,999	3.9526
	4.4 mm	2,351	3.0303
	4.9 mm	1,830	2.6738
	5.4 mm	1,465	2.3923
	5.9 mm	749.6	2.2
16 mm	2.4 mm	59,369	12.987
	2.9 mm	35,915	10.101
	3.4 mm	24,042	8.2645
	3.9 mm	17,214	6.993
	4.4 mm	10,066	5.3476
	4.9 mm	8,058	4.7847
	5.4 mm	5,051	3.7879
	5.9 mm	3,991	3.367
	6.9 mm	3,027	2.9326
	8.4 mm	2,245	2.5253
	10.1 mm	1,263	1.8939
	12.5 mm	1,036	1.7153
25 mm	1.4 mm	21,157	9.0909
	1.9 mm	14,692	7.5758
	6.1 mm	3,385	3.6364
	10.1 mm	1,199	2.1645
	16.2 mm	608	1.5408
	22.3 mm	444	1.3175

4 Image and 3D Data Processing Pipeline

4.1 Texture based Solder Paste Detection

4.1.1 General

The detection and segmentation of solder paste is an essential processing step, when measurements of the solder paste (extent, area, volume, etc.) are required. One specific part of the system are the two LED bar lights which are used to illuminate the PCB at a low angle from two directions (comparable to a dark field illumination). Since no diffuse and omnidirectional illumination is used the roughness of the grainy solder paste becomes visible in the image. This appearance differs from all other components of the PCB (board itself, solder pads, writings, lines, etc.) and can be used to identify segments of solder paste. Therefore, a structure based analysis of pre-segmented regions is used.

4.1.2 Algorithm

The texture parameters of Haralick (Haralick features) are used to differentiate between solder paste regions and all other segmented regions. Previously, an image pre-processing and segmentation is needed to extract the Haralick features from segmented regions, as shown in the flow chart in figure 4.1. In the first step the images, captured by the camera, are pushed into the processing chain. The pre-processing of the images contains a median filtering to remove small dust particles and a gradient filtering (see equation 4.1 and 4.2) to enhance the structure of solder paste regions.

$$grad(f(x,y)) = \nabla f(x,y) = \begin{pmatrix} \frac{\partial f(x,y)}{\partial x} \\ \frac{\partial f(x,y)}{\partial y} \end{pmatrix} = \begin{pmatrix} f\prime_x(x,y) \\ f\prime_y(x,y) \end{pmatrix} \tag{4.1}$$

$$|grad(f(x,y))| = \sqrt{f\prime^2_x(x,y) + f\prime^2_y(x,y)} \tag{4.2}$$

Afterwards the image is binarized using Otsu's thresholding method. For each segmented region the bounding box is computed and eventually emerged holes in the regions are filled by a flood fill from the upper left corner of the region, inversion of the result and bitwise or operation of the thresholded image and the inverted flood filled image. For every extracted region (solder and non-solder), up to 4 co-occurrence matrices, using the displacement vectors presented in figure 2.15 and the schematic from figure 2.16, were used to compute the 14 Haralick features. The obtained parameters were examined to determine the best parameters for the most feasible classification of solder joints and non solder regions. As it will be shown in the results, the best parameters to differentiate between solder and non-solder regions are *contrast, variance, homogeneity, energy*

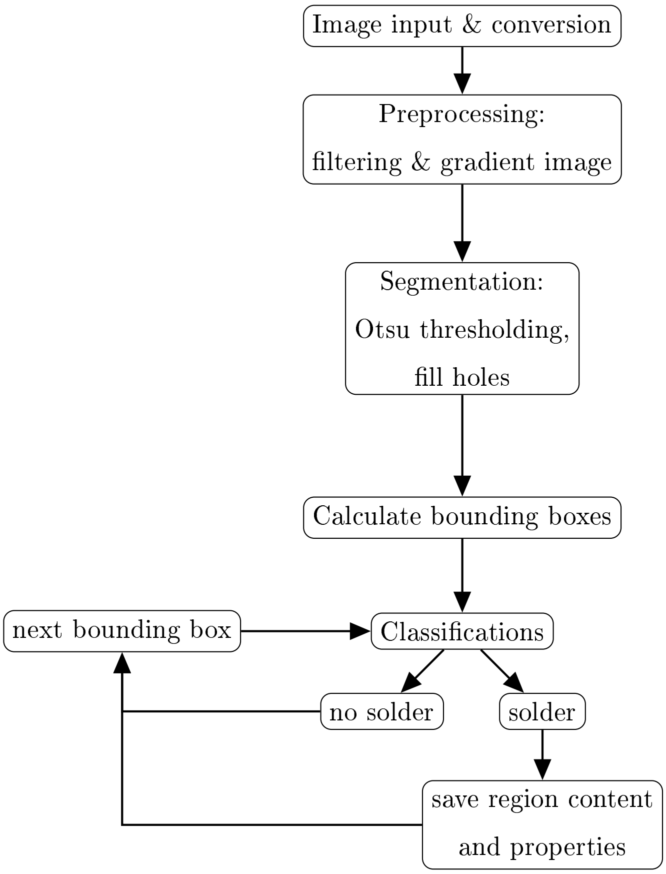


Figure 4.1 Flow chart of texture based solder paste detection

and entropy, which are the only parameters to be computed in the final processing chain. Other parameters deliver redundant or improper classifications.

$$GLCM_1 \rightarrow f_{contrast_1}; f_{variance_1}; f_{energy_1}; f_{homogeneity_1}; f_{entropy_1} \tag{4.3}$$

$$GLCM_2 \rightarrow f_{contrast_2}; f_{variance_2}; f_{energy_2}; f_{homogeneity_2}; f_{entropy_2} \tag{4.4}$$

$$GLCM_3 \rightarrow f_{contrast_3}; f_{variance_3}; f_{energy_3}; f_{homogeneity_3}; f_{entropy_3} \tag{4.5}$$

$$GLCM_4 \rightarrow f_{contrast_4}; f_{variance_4}; f_{energy_4}; f_{homogeneity_4}; f_{entropy_4} \tag{4.6}$$

Afterwards the mean value for every texture parameter is computed from the four values. Thus, for each region five mean texture values exist.

$$\bar{f}_{contrast} = \frac{f_{contrast_1} + f_{contrast_2} + f_{contrast_3} + f_{contrast_4}}{4} \tag{4.7}$$

$$\bar{f}_{variance} = \frac{f_{variance_1} + f_{variance_2} + f_{variance_3} + f_{variance_4}}{4} \tag{4.8}$$

$$\bar{f}_{energy} = \frac{f_{energy_1} + f_{energy_2} + f_{energy_3} + f_{energy_4}}{4} \tag{4.9}$$

$$\bar{f}_{homogeneity} = \frac{f_{homogeneity_1} + f_{homogeneity_2} + f_{homogeneity_3} + f_{homogeneity_4}}{4} \tag{4.10}$$

$$\bar{f}_{entropy} = \frac{f_{entropy_1} + f_{entropy_2} + f_{entropy_3} + f_{entropy_4}}{4} \tag{4.11}$$

Within a training set of PCB images, thresholds for parameters with a significant separation of solder paste and non-solder areas are setup to perform the classifications for new images. A region is classified if the computed relevant Haralick features are above, or below respectively, the determined threshold, depending on whether the values of the solder joints for a specific parameter are smaller or larger than the non-solder regions.

4.1.3 Evaluation

Figure 4.2 shows an example image of a PCB section with marked solder and non-solder areas. An exemplary overview of these areas is given in figure 4.3.

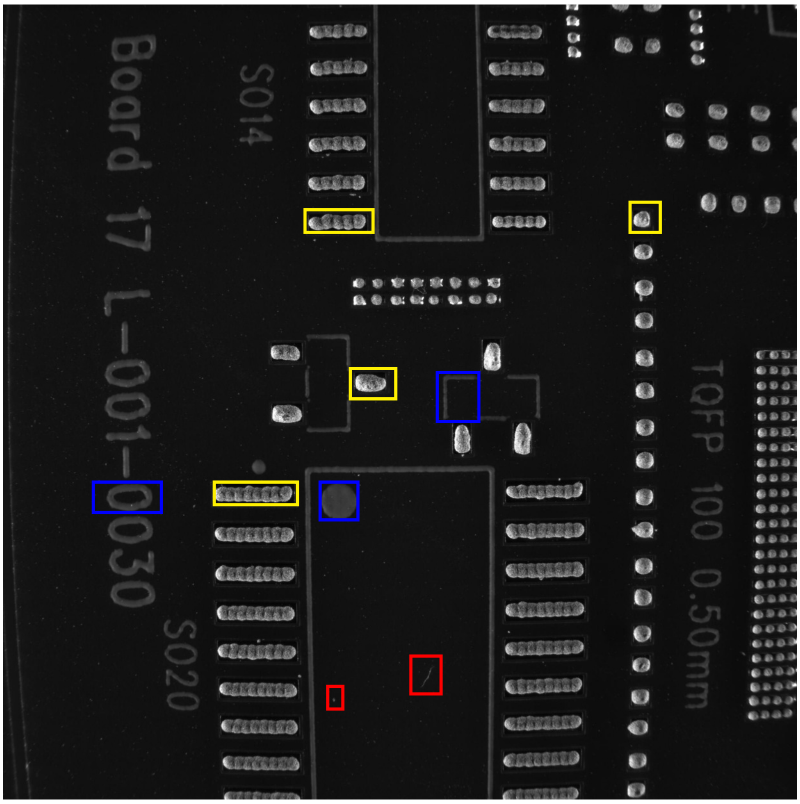


Figure 4.2 Structures of a PCB: yellow: solder paste, blue: frames and writings, red: dust and other pollution.

The developed solder paste detection method was tested using three different data sets of the test PCB:

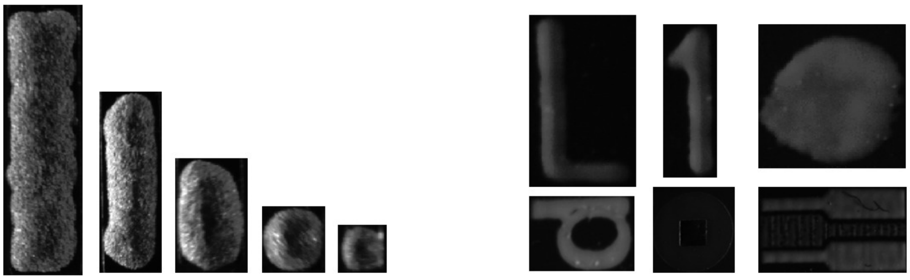


Figure 4.3 Overview of different solder joints (right) and non-solder-objects (left)

1. strong polluted surface, one-sided illumination
2. clean surface, one-sided illumination
3. clean surface, double-sided illumination (final setup)

Examples of the data sets are shown in figure 4.4. The evaluation of the solder paste detection was done by the examination of the sensitivity, detection rate and positive predictive value (PPV). Therefore, each classification is evaluated using the truth matrix, represented by table 4.1. The detection rate indicates the precision of the classification. If the detector runs on a data set with two classes (solder and non-solder) the truth matrix can be set up. The classification decision can be divided into four cases:

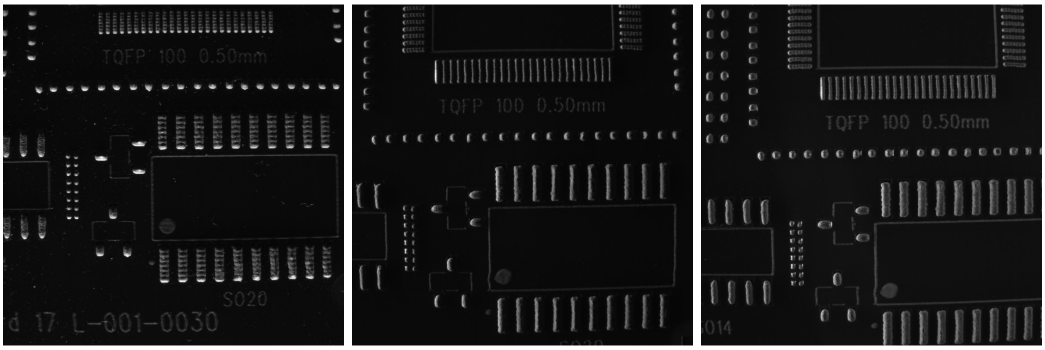


Figure 4.4 Example images of data set 1 (left), 2 (center) and 3 (right)

Table 4.1 Truth matrix of a binary classification

	solder paste present	no solder paste present
solder paste detected	true positive	false positive
no solder paste detected	false negative	true negative

- **true positive (TP)**: correctly detected solder paste; solder paste was detected were solder paste is present

- **true negative (TN)**: no solder paste detected; in regions were no solder paste is present, no paste is detected
- **false positive (FP)**: wrong detection of solder paste; solder paste is detected in regions, where no solder paste is present
- **false negative (FN)**: missing solder paste; no solder paste is detected in regions containing solder paste

From these cases the sensitivity and PPV is calculated as follows[Toe12]

$$sensitivity = \frac{TP}{TP + FN} \tag{4.12}$$

$$PPV = \frac{TP}{TP + FP} \tag{4.13}$$

Every examined region was also classified manually to obtain the ground truth.

4.2 Feature Detection and Matching

4.2.1 General

Over the past years many feature detectors have been developed. Dependent on the application each detector delivers differing results and so the subsequent image feature matching does [LRKB13]. In terms of solder joint inspection, it is important to obtain as many features on solder joints and as less features on the background (PCB). Moreover, the feature points should be uniformly spaced within a solder joint area, to assure a suitable 3D reconstruction. For the setup of a suitable processing 6 different feature detectors are implemented, examined and compared against each other. For this evaluation a new parameter to asses the result of the feature detectors in terms of applicability for solder joint key point detection is introduced. The parameter is called *solder joint feature coverage (SolFeC)*. Prior to the reconstruction, an indispensable step is the feature matching as well as the elimination of false matches, which occur in large numbers due to the homogeneous appearance of solder joints and thus to a short vector distance of different feature descriptors that do not belong to corresponding feature points. Therefore, a vector distance as well as an epipolar geometry based matching correction are examined and a novel model based correction approach is proposed. In this thesis the performance of feature detection and matching methods in terms of solder joint features is evaluated for the first time. Therefore, novel parameters and methods were developed and evaluated in addition to common evaluation parameters like, computation time, recall and precision.

4.2.2 Datasets

Images of the examined PCB were acquired using a specific range of image brightness, to provide a proper imaging that enhances the visual structure of the solder joints without saturated image regions. These variations lead to data sets which are visualized exemplary in part in figure 4.5. The examined data set holds 10 PCBs with different solder paste amounts and 30 illuminations resulting in 300 PCB acquisitions. All previously considered feature detector and descriptor algorithms are implemented, varied by their parameter settings and examined to their capability of solder joint feature detection.

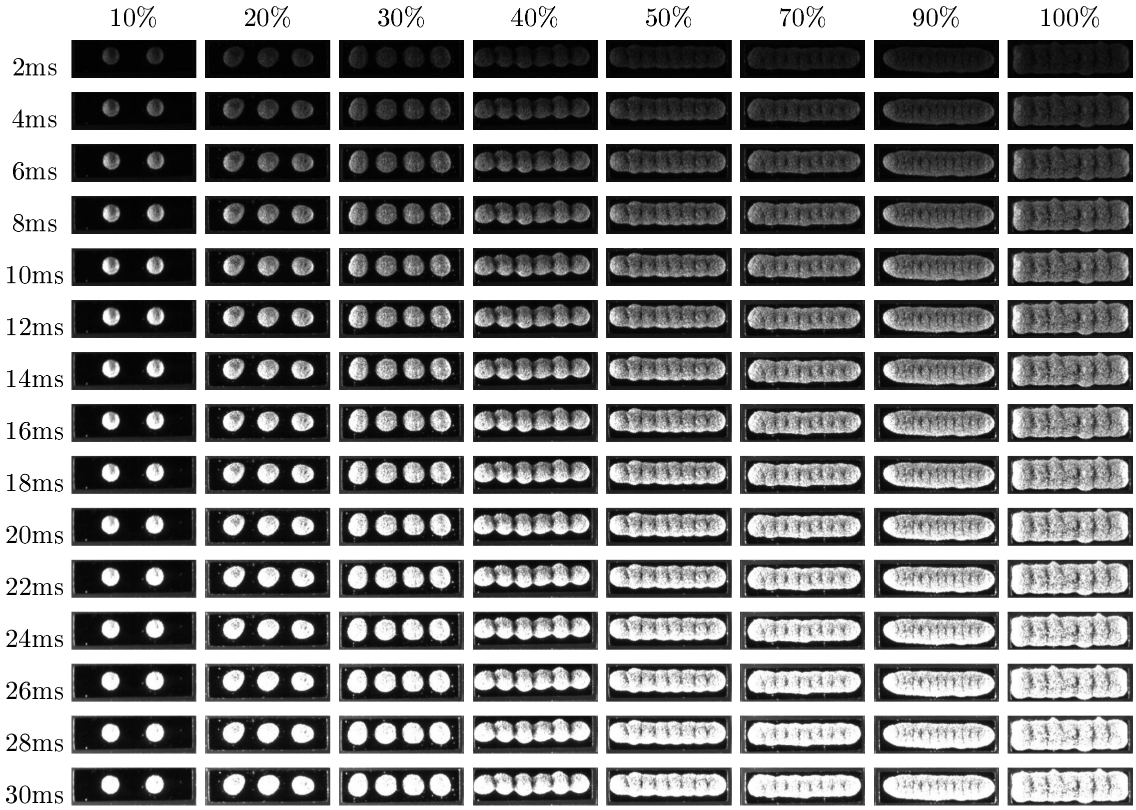


Figure 4.5 Exemplary acquisition of solder joints: The test solder joints yielded a coverage of the solder pads from 10% to 100% (left to right). A proper illumination is determined by adjusting the cameras integration time from low values, where the solder joint is rarely visible to values where the joints are within the cameras saturation (top to bottom). This is done to obtain a proper image contrast using the corresponding integration time for an ideal image acquisition in terms of feature extraction.

4.2.3 Detection and Description

Besides the introduced feature detectors the performance of ORB and FAST was evaluated using openCV's CUDA implementation for these detectors. For every illumination condition the parameter configurations as indicated in table 4.2 were changed. After every detector parametrization and utilization the available descriptors (SIFT, SURF, BRIEF, ORB, BRISK) were applied prior to the feature matching.

Table 4.2 Parameter configuration of feature detectors

Detector	Parameter	Range
SIFT	number of octaves	1 - 5
	edge threshold	20,10,5,3,1
	contrast threshold	0.01 - 0.05
SURF	number of octaves	1 - 5
	layers per octave	20, 10, 5, 3
	hessian threshold	500, 300, 100, 1
FAST	contrast threshold	1 - 50
	hardware	CPU, GPU
ORB	pyramid levels	0 - 8
	contrast threshold	1 - 50
	hardware	CPU, GPU
BRISK	Contrast Threshold	1 - 50
	number of octaves	1 - 10
MSER	delta gray level	1 - 10

4.2.4 Correction of false Matches

Vector Distance based Correction Usually the matches of feature points are established by comparing the distance (see section 2.4.7.2) between the corresponding feature descriptors (point pairs). Since the image content around a feature point might change by rotation, scale, illumination, etc., the descriptor vector of a feature point in the first image is not exactly the same compared to the descriptor vector in the second image. Thus, a threshold t_{dist} is used to determine if feature points and their descriptors, respectively, belong to each other. Therefore, the vector distance between the two feature point descriptor vectors $\vec{a}$ and $\vec{b}$ is calculated and a match is established if the distance is below the user defined threshold t_{dist} [HZ03]:

$$pp_{match}\left(\vec{a},\vec{b}\right) = \begin{cases} true\ ;\ if\ d\left(\vec{a},\vec{b}\right) < t_{dist} \\ false\ ;\ otherwise \end{cases} \tag{4.14}$$

Fundamental Matrix and Epipolar Geometry based Correction Besides the simple method of comparing descriptor distances, the epipolar geometry and the corresponding fundamental matrix can be considered to eliminate remaining false matches. The epipolar geometry describes the geometric relation of the two cameras and the epipolar plane, which includes the base line (the line between the camera centers C and a considered 3D world point X. The baseline also consists of the epipoles $\boldsymbol{e}$ and $\boldsymbol{e}'$, which are projections of the camera center of camera 1 onto the image plane of camera 2 and vice versa (see figure 4.6). Points $\boldsymbol{X}$, $\boldsymbol{C}_1$, $\boldsymbol{C}_2$, $\boldsymbol{x}$, $\boldsymbol{x}'$, $\boldsymbol{e}$ and $\boldsymbol{e}'$

are coplanar respectively. This condition is called the coplanarity constraint. This means a corresponding point $\boldsymbol{x}'$ of $\boldsymbol{x}$ must lie on the so-called epipolar line, which is defined by the intersection of the epipolar plane and the image sensor plane. Thus, there is a mapping:

$$\boldsymbol{x} \mapsto \boldsymbol{l}'$$

This projective mapping from points to lines is defined by the fundamental matrix F.

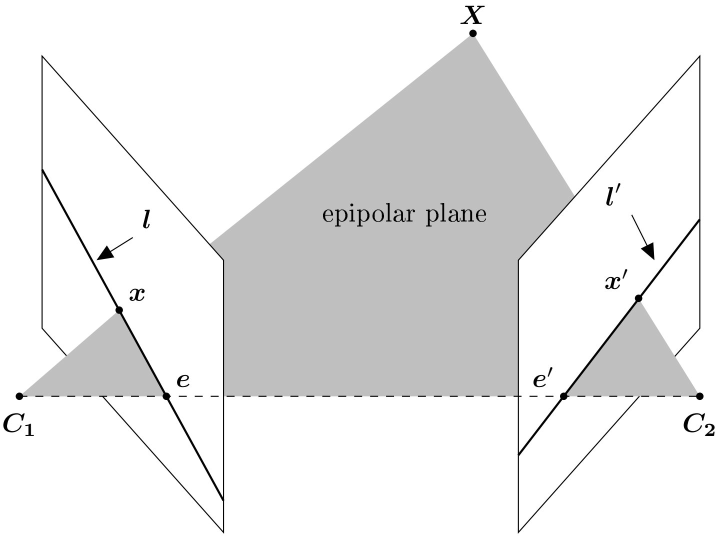


Figure 4.6 Epipolar geometry: C_1 and C_2 are the camera centers, a world point $\boldsymbol{X}$ is projected within the epipolar plane into the images and results in the image points $\boldsymbol{x}$ and $\boldsymbol{x}'$. The projection of one camera center into the other camera results in epipoles $\boldsymbol{e}$ and $\boldsymbol{e}'$. $\boldsymbol{e}$ and $\boldsymbol{x}$ as well as $\boldsymbol{e}'$ and $\boldsymbol{x}'$ setup the epipolar lines $\boldsymbol{l}$ and $\boldsymbol{l}'$ within the image planes [cf. HZ03].

Consequently, if a point $\boldsymbol{x}'$ does not follow the coplanarity constraint, this point cannot establish a match to the point $\boldsymbol{x}$. Since a calibrated camera is used the fundamental matrix is obtained by equation 4.15.

$$F = [P'\boldsymbol{C}] \times P'P^{+} \tag{4.15}$$

In this equation P^{+} is the pseudo inverse of P ($PP^{+} = I$), I is the identity matrix, P' is the camera of the second camera and C is the camera center of the first camera. Thus, $[P'C]$ is the projection of the camera center C of the first camera onto the image plane of the second camera via its projection matrix P' and defines the epipole $\boldsymbol{e}' = [P'\boldsymbol{C}]$. A correct point correspondence will meet the following condition:

$$\boldsymbol{x}'F\boldsymbol{x} = 0 \tag{4.16}$$

Due to the photogrammetric normal case setup and the translational motion parallel to the image plane an image point and its correspondence satisfies the condition by the fundamental matrix but so do false matches on the mapped line $\boldsymbol{l}'$. The intersection of the baseline and the image sensor plane is at infinity and thus the epipoles lie at infinity. Using these considerations the match is evaluated with [HZ03]:

$$pp_{match}\left(\vec{a},\vec{b}\right) = \begin{cases} true\ ;\ \boldsymbol{x}'F\boldsymbol{x} = 0 \\ false\ ;\ otherwise \end{cases} \tag{4.17}$$

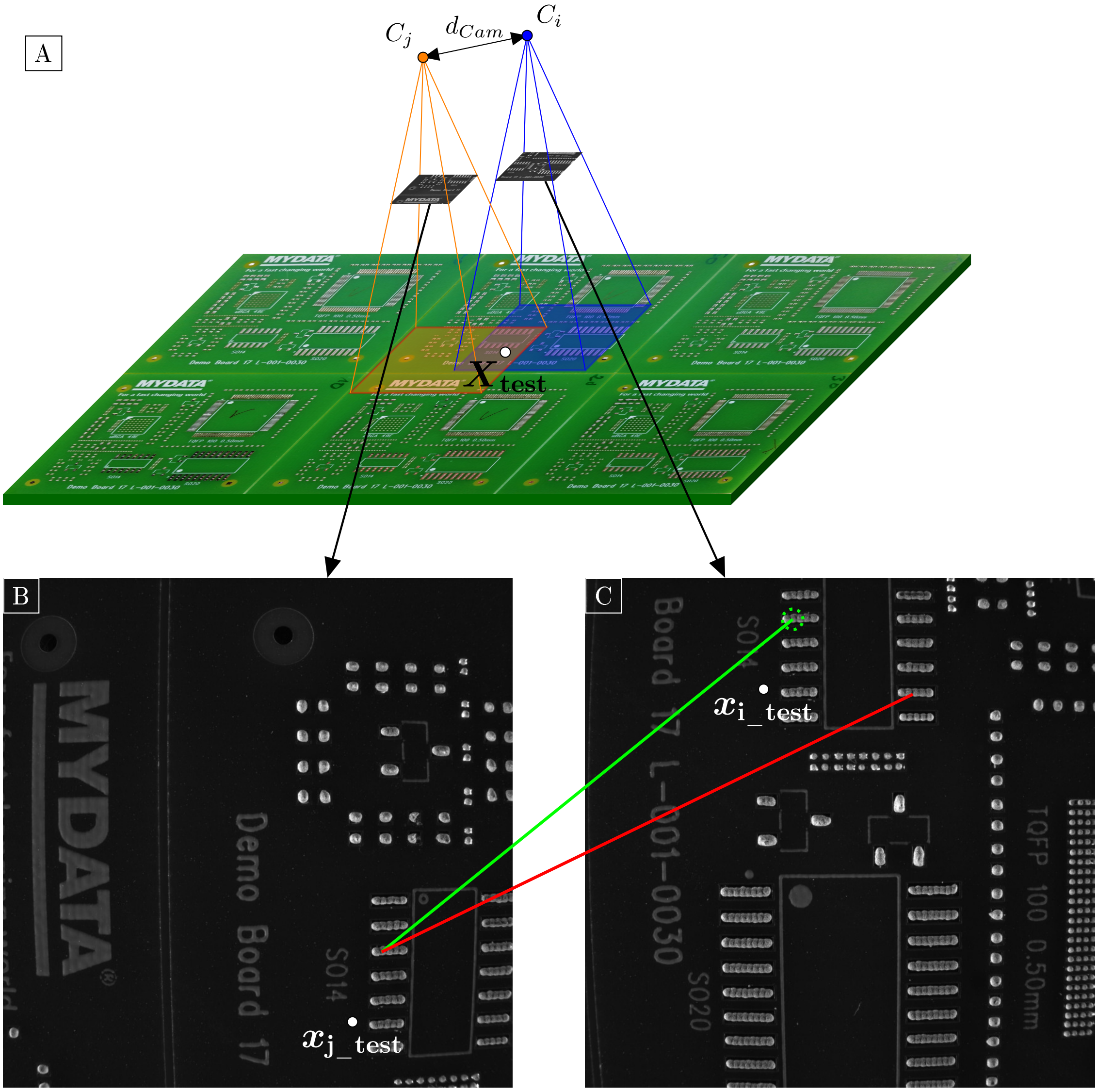


image at camera position C_j image at camera position C_i

Figure 4.7 Camera model based matching correction: A virtual test point $\boldsymbol{X}_{test}$ is considered within the overlapping field of views (orange and blue) of the two camera positions. The point is set at $d_{Cam}/2$ with a height corresponding to the PCB surface. The point is projected into the camera images at positions i and j $(i \neq j)$ resulting in the points $\boldsymbol{x}_{i\_test}$ and $\boldsymbol{x}_{j\_test}$. From these points a feature shift is computed. A correct match is indicated by the green line and a false match by the red line. The green dashed circle in C indicates the tolerance circle of radius s_{tol} (not to scale). Since the second (red) match does not lie within that circle, it is considered to be false.

Camera Model based Correction Figure 4.7 A shows the acquisition position of two PCB areas (red and blue) and the resulting images in figure 4.7 B and C. A correct and a false match is shown in the image pair in figure 4.7 B and C in green (correct) and red (false). Due to the specific application field of *ultra-close range normal case photogrammetry*, the known camera models and the restricted object geometry can be used as an a-priori knowledge to eliminate false matches. Here, restricted object geometry means, that the PCB is almost flat, showing a minimal warpage and the solder joint heights are within the very short depth of field of 800 μm. The efficient way to correct the set of matches is the evaluation of the pixel shift a feature point gets by camera motion. Therefore, an expected shift in x- and y-direction (x_{shift} and y_{shift}) needs to be computed taking the lens distortion into account (see equation 4.20). An arbitrary 3D point $\boldsymbol{X_{test}}$ at the surface of the PCB is projected onto the image planes using the calibrated camera projection matrices P_i and P_j, resulting in the image points $\boldsymbol{x}_{i\_test}$ and $\boldsymbol{x}_{j\_test}$. In general, the point $\boldsymbol{X_{test}}$ was chosen to be in-between the two camera centers C_i and C_j) at the height of the PCB (PCB_h) (see equation 4.18).

$$\boldsymbol{X}_{test} = \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = \begin{bmatrix} (C_{iX} + C_{jX})/2 \\ (C_{iY} + C_{jY})/2 \\ PCB_h \\ 1 \end{bmatrix} \tag{4.18}$$

$$x_{i\_test} = P_i \boldsymbol{X}_{test} \text{ and } x_{j\_test} = P_j \boldsymbol{X}_{test} \tag{4.19}$$

$$\begin{bmatrix} x_{shift} \\ y_{shift} \end{bmatrix} = \begin{bmatrix} \boldsymbol{x}_{i\_test}(x) - \boldsymbol{x}_{j\_test}(x) \\ \boldsymbol{x}_{i\_test}(y) - \boldsymbol{x}_{j\_test}(y) \end{bmatrix} \tag{4.20}$$

Note, that not the absolute value of the distance is used, since the shift direction is also of importance and other false matches may result with a similar absolute distance. Since the height of the currently evaluated 3D point is not known, the shift is allowed to vary within a user defined circular region with radius s_{tol} to allow for higher 3D points, that result in different image point shift than points at the PCB surface. For a detected feature point in the first image a predicted feature point and match candidate is obtained by adding the expected shift values to its coordinates. The actual shift of the matched feature points $\boldsymbol{x}_{fi}$ and $\boldsymbol{x}_{fj}$ is also calculated using equation 4.20 and a decision is made:

$$pp_{match}\left(\vec{a}, \vec{b}\right) = \begin{cases} true\ ; & \boldsymbol{x}_{fi}(x) - \boldsymbol{x}_{fj}(x) = x_{shift} \pm s_{tol} \text{ and} \\ & \boldsymbol{x}_{fi}(y) - \boldsymbol{x}_{fj}(y) = y_{shift} \pm s_{tol} \\ false\ ;\ otherwise & \end{cases} \tag{4.21}$$

Here i and j are the indices of the feature within the two images that are considered for the current matching step.

4.2.5 Evaluation

4.2.5.1 Feature Detection

Subjective Evaluation: In the first evaluation process the results of the feature detection and its parametrization was done visually by subjectively estimating the homogeneity, coverage and distribution of the feature points over a solder joint. However, this cannot give an objective statement of the detection result.

Objective Evaluation: Besides the number of obtained feature points, computation time and if the feature detector algorithm is free available a novel objective measure is introduced to evaluate the feature detection in terms of distribution, coverage and homogeneity. The novel parameter is called the *solder joint feature coverage* (*SolFeC*), which consists of 5 components, that take the following inputs and properties into account:

- segmented solder joint
- geometric properties of the segmented area as explained in section 2.4.2
- 2D feature point coordinates
- feature point-to-point distances

The solder joints are segmented automatically via Otsu's thresholding and classified by the proposed texture based analysis to obtain a coherent solder joint region. Every keypoint coordinate $pt_f = (x_f, y_f)$ is assigned to the corresponding segmented solder joint region resulting in keypoint clusters. For each cluster the point-to-point euclidean distances $d_{ptp\ ij}$ between every feature point i and j, $(i, j \in \mathbb{N} | 1 \leq i, j \leq n_f)$ is calculated, where i and j are the feature indices and n_f is the number of features within the corresponding solder joint:

$$d_{ptp\ ij} = \sqrt{(x_{fi} - x_{fj})^2 + (y_{fi} - y_{fj})^2} \tag{4.22}$$

If a solder joint is covered properly by feature points, the maximum distance between two feature points $\max d_{ptp\ ij}$ is close to the length of the maximum line l_t that can occur within the region. Thus, the first component, the relation d_{rel} between these values is close to one. The maximum length is determined by the region's minor and major axis length: (l_{mi} and l_{ma}): $l_t = \sqrt{l_{mi}^2 + l_{ma}^2}$.

$$d_{rel} = \max d_{ptp\ ij} / l_t \tag{4.23}$$

The next component for the evaluation is the centroid shift c_{shift}. The first centroid $c_r = (x_{c_r}, y_{c_r})$ results from the segmentation of every single solder joint (region centroid), whereas the second centroid $c_f = (x_{c_f}, y_{c_f})$ is setup from all feature points that occur within that segmented solder joint. Afterwards, the euclidean distance between the centroids c_r and c_f is calculated. The value becomes

$$c_{shift} = \sqrt{\left(x_{c_f} - x_{c_r}\right)^2 + \left(y_{c_f} - y_{c_r}\right)^2} \Bigg| c_f = \frac{1}{n_f} \sum_{i=1}^{n_f} pt_{fi} \tag{4.24}$$

If the distribution of feature points does not deviate from a central distribution the shift between the two centroids will become zero. Since the shift is measured in pixels it is scaled by the major axis length l_{ma} and subtracted from one to obtain zero when the maximum shift is reached and one if the shift is zero. Additionally to the distances and centroid shift components a solder joint is considered to be covered properly by features if the mean distance between feature is less or equal to one-tenth of the regions minor axis length. Therefore, a second distance component is calculated. All feature points are triangulated by the two-dimensional Delaunay triangulation [Wat81]. From the resulting triangles the edge lengths e are extracted. From these edge lengths the number of all edges n_{le} with a length larger than one-tenth of the regions minor axis length are extracted and the mean length is calculated and set into relation to the regions major axis length:

$$d_{le} = \frac{\sum_{i=1}^{n_{le}} le}{n_{le}} / l_{ma} \; |le = e \text{ if } e > \frac{l_{mi}}{10} \tag{4.25}$$

d_{le} becomes one if the mean value of the long edges becomes zero. That means, if all features are close enough to each other, the corresponding edge length is smaller than a tenth of the minor axis length. If not, that part of the equation will decrease to zero. The fourth factor takes the mean distances (edge lengths) between the feature points into account. If this value is small enough this component will be set to one otherwise to the relation between the mean value of the distances in relation to the tenth of the minor axis length:

$$w_d = \begin{cases} 1 & \text{if } \frac{10 \cdot \bar{e}}{l_{mi}} < 1 \\ \frac{10 \cdot \bar{e}}{l_{mi}} & \text{else} \end{cases} \tag{4.26}$$

Finally, a component A_{rel} describes, if the feature points are spread over the currently considered solder joint region. Therefore, the concave hull of the feature points is computed and its enclosed pixel area is related to the number of pixels of the segmented solder joint region.

$$A_{rel} = \frac{\text{number of solder paste pixels}}{\text{number of pixels in the filled concave hull}} \tag{4.27}$$

This component becomes one if the relation between the area within the features concave hull and the segmented pixel area are equal. A badly covered solder joint will show a smaller feature based concave hull and thus, this component will decrease. Putting all 5 components together, a quantitative measure of the solder joint coverage is obtained:

$$SolFeC = \frac{d_{rel} + (1 - (c_{shift}/l_{ma})) + (1 - d_{le}) + 1/w_d + A_{rel}}{5} \tag{4.28}$$

Using formula 4.28 the coverage of a solder joint with features can be described. Each component ranges between 0 and 1, where every summand in equation 4.28 becomes one for a perfect coverage and so the SolFeC-parameter does.

4.2.5.2 Feature Matching

The obtained matches are evaluated by the recall, precision and 1-precision parameters, which are commonly established parameters to describe the performance of a feature matching. Besides these performance parameters also the computation time of the CPU-based brute force, FLANN and GPU-based brute force matcher, provided by openCV is evaluated.

Recall The recall parameter is defined by the ratio of correct positive matches and the number of correspondences and thus all possible matches. It describes how many of possible matches have been found.

$$recall = \frac{number\ of\ correct\ positives}{total\ number\ of\ correspondences} \tag{4.29}$$

Precision Precision is a measure that considers the correct positives over all matches, no matter if they are correct or false. Thus, a low number of false matches will result in a high precision value.

$$precision = \frac{number\ of\ correct\ positives}{total\ number\ of\ matches\ correct\ or\ false} \tag{4.30}$$

1-precision The 1-precision parameter in turn compares the number of false positive matches to the total number of found matches. A low value for this parameter means good performance of the feature matching method. Thus, it is basically the inverse of the precision measure.

$$1 - precision = \frac{number\ of\ false\ positives}{total\ number\ of\ matches\ correct\ or\ false} \tag{4.31}$$

The number of correspondences (possible matches) is the number of keypoints in the overlapping areas of the considered images. Correct positives define matches of keypoints that really belong together while false positives are matches that connect keypoints which do not correspond to each other [TSH^{+}10]. As equations 4.29 to 4.31 show, numbers like the amount of established matches and the number of possible matches are needed for the calculation of the evaluation scores. Usually the number of possible matches corresponds to the number of detected image features. Due to the photogrammetric normal case these numbers were adjusted. Overlapping image areas are calculated by projecting the edges of an image onto the PCB plane and further into the second camera. Thus, feature points outside this area cannot establish a match between the two images and should not be taken into account for the precision-recall calculation. The number of correct matches can be obtained by manual counting. However, due to the large amount of features and matches (>10k) this would lead to a non-acceptable effort. Thus, the number of correct matches is obtained by the camera model based matching correction from section 4.2.4. The success of this method is finally shown by the subsequent reconstruction.

4.3 3D Reconstruction and Point Cloud Processing

4.3.1 General

After a successful detection and matching of image feature points the modified camera models for each image can be used to start the 3D processing. The pipeline of this part contains the steps as visualized in the flow chart in figure 4.8. Triangulation is performed using the direct linear triangulation as explained in section 2.4.8.

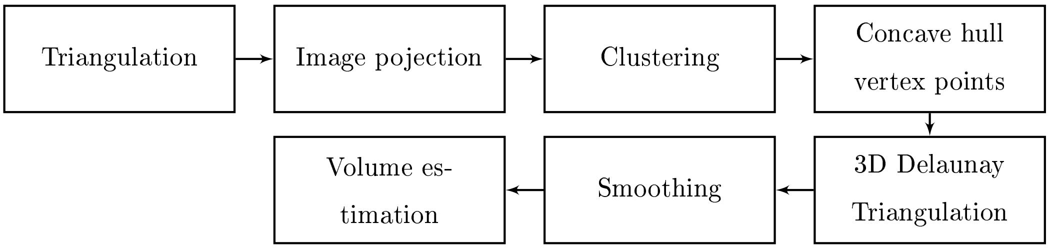


Figure 4.8 Flowchart of the 3D processing.

4.3.2 Clustering

The resulting point cloud needs to be processed to obtain the solder joints and the corresponding volumes. First of all, a separation of the point cloud into solder joint related clusters is necessary. This clustering was implemented in two ways. The first option is performed imaged based during the feature detection (as described in section 4.2.5.1). The second option is based on point distances of the reconstructed point cloud using euclidean clustering [RUS09]. However, due to inhomogeneity of the point cloud, depending on its point density, the clustering can lead to inconsistent results. A main factor of the clustering is the distance parameter, which determines if a point is connected to a considered cluster. If the triangulated 3D points of a solder joint are farther apart than the distance to a neighboring solder joint, points of different solder joint become connected and points within the same solder joint my become disconnected, resulting in false clusters. The cloud is improved by projecting the acquired and segmented images onto the PCB plane to overcome the problem of false clustering results, which corresponds to the PCB surface. Thus, the triangulated points obtain an underlying solder joint region. Since the PCB plane is considered with $z = 0$ the projection becomes a homography with the homography matrix H, which can be derived from the projection matrix P:

$$H = \begin{pmatrix} p_{11} & p_{12} & p_{14} \\ p_{21} & p_{22} & p_{24} \\ p_{31} & p_{32} & p_{34} \end{pmatrix} \tag{4.32}$$

For the transform from the image plane to the PCB plane the inverse homography is used. Thus an image point $\boldsymbol{x}$ (pixel coordinates) is transformed to a 3D world point $\boldsymbol{X}$ ($z = 0$) by:

$$\boldsymbol{X} = H^{-1}\boldsymbol{x} \tag{4.33}$$

Using this additional information reduces the clustering problem from three to two dimensions. For the purpose of clustering the z-coordinate of the triangulated points is considered to be zero too and the clustering is performed in x- and y- direction. The cloud to be processed is composed of the triangulated point cloud and the point cloud resulting from the inverse projection of images onto the PCB plane. Subsequently, for every point in the cloud the set of neighboring points within a sphere of radius r, with $r < d_{max}$, is searched. d_{max} is the maximum distance two points can have to belong to the same cluster. Since the projected images result in a dense zero-level point cloud under the comparable sparse 3D point cloud, the clustering distance can be set to the distance of the points within the zero-level cloud. This leads to perfectly separated solder joint clusters, whereas a unique cluster for 3D points of the same solder joints is preserved.

4.3.3 Surface Reconstruction and Volume Estimation

Due to processing parts of inter and extrapolation in surface reconstruction algorithms like Poisson surface reconstruction [KBH06] or the necessity of a dense and homogeneous point cloud when using the fast greedy triangulation algorithm [dBCvKO08, p. 55 - 58], these and similar algorithms are not appropriate for a surface reconstruction of the irregular solder joint point clouds obtained from ultra-close range normal case photogrammetry. Additionally the cloud can contain points under the surface (e.g. due to small feature point localization errors), which yields unfeasible and non-closed surfaces from the mentioned algorithms. For the final volume calculation a closed surface for each solder joint is indispensable. The first step for this was done by the clustering. Thus, the following steps are performed for every obtained cluster representing a solder joint.

Concave Hull From the clusters, containing projected ground points and the reconstructed 3D points a concave hull is constructed to obtain a surrounding hull of a solder joint to compute its volume from. A concave hull is also called an α-shape [EM94]. For a cluster $i \mid i = [1, N]$ (N - number of clusters), the set of points $S_i \mid S \in \mathbb{R}^3$ is considered. Based on the α parameter, different shapes are obtained for a point set, since this parameter determines, which point distances can establish a connection. If $\alpha \to 0$ the shape approaches single points corresponding the original point set. If $\alpha \to \infty$ the resulting hull approaches the convex hull of the point set. The generation of an alpha shape is based on the Delaunay triangulation. A simplex of the delaunay triangulation is part of the alpha shape if the circumcircle of a subset T ($|T| = 4$; 4 vertex points to build tetrahedron simplex) with radius $r < \alpha$ is empty or if the simplex is a boundary simplex. The algorithm to compute a concave hull was implemented using the Point Cloud Library (PCL).

Unfortunately this library does not take care about normal orientation of the resulting triangles. This leads to triangles with a normal pointing to the inside of a solder joint, resulting in a non closed surface in a computer graphics point of view.

VTK Delaunay Triangulation Only the vertex points of the obtained triangles are used for the further processing to overcome the mentioned problem of misoriented triangles. This point cloud containing only concave vertex points is given to the Delaunay triangulation implemented in the Visualization Toolkit (VTK), which ensures a proper alignment of the resulting surface triangles [SML06].

Surface Filtering A surface smoothing filter is applied to remove the sharp edges due to outliers and to obtain a smooth surface. For every point in the cloud of vertex points v, connected vertex points are calculated. This can be accelerated by using the obtained Delaunay triangles. After the setup of the connectivity list an iterative loop is started. For each point v, the average of the connected vertices is computed and the considered vertex is moved to that position. Via a relaxation factor the movement of the vertex can be controlled to suppress a to strong smoothing. The only sharp edge that should preserve is the edge between solder joint surface and the projected ground points. Otherwise this edge will move into the inside of the original solder joint. Thus, only vertex points with a z-coordinate > 0 are considered [SML06].

Volume Estimation After a closed surface of a solder joint is obtained its volume can be estimated using equations 4.35 and 4.34 [ALDF94]. At first the volumes are divided in cubes by the dividing cubes algorithm. It can be considered as a sampling of the volume with large cubes inside the volume and a subsampling with smaller cubes at the surface [CLL$^+$88]. This algorithm results in a point list that is a 3D surface representation of the solder joint. For each point the maximum unit normal component of the unit normal vectors are calculated.

$$volume = k_x \sum_i (x_i n_{x_i} \Delta a_i) + k_y \sum_i (y_i n_{y_i} \Delta a_i) + k_z \sum_i (z_i n_{z_i} \Delta a_i) \tag{4.34}$$

$$\Delta a_i = \begin{cases} \frac{\Delta x \Delta y}{|n_{z_i}|} & \text{if } n_z \text{ is the maximum unit normal component,} \\ \frac{\Delta x \Delta z}{|n_{y_i}|} & \text{if } n_y \text{ is the maximum unit normal component,} \\ \frac{\Delta y \Delta z}{|n_{x_i}|} & \text{if } n_x \text{ is the maximum unit normal component} \end{cases} \tag{4.35}$$

The algorithm of these equations is called divergence theorem algorithm. The volume of a solder joint is estimated from the corresponding point list. x, y, z are point coordinates and k_x, k_y, k_z are weighting coefficients whose sum is equal to one. Δa_i is given by equation 4.35. These factors are calculated by the sum of points with the maximum unit normal component pointing in the direction of the corresponding index and is divided by the total number of points. Each part of 4.34 calculates the entire volume. The volumes are weighted by the k-factors which results in a more precise volume estimation.

5 Demonstrator - Design, Setup and Software

5.1 General

For the proof of principle as well as for examinations of the single processing steps a demonstrator system with a corresponding software was developed. The hardware design and setup has the capability of acquiring proper illuminated images of solder paste printed PCBs. Due to the intended application of a camera at the printer head, a motion system has to perform the translation of the camera. Not indispensable for the demonstrator but for an integration into the printer a high frame rate of the camera with a fast data transfer to the image processing system is needed. A proper lightning system is required to save weight that is put onto the printer and to ensure a unique illumination for the entire printing area. A software framework is developed to encapsulate all processing steps and the control of all hardware components . The software "SolderInspect" developed in this thesis was programmed under Linux using mainly C++ to facilitate a transfer ro the printer. Test PCBs for examination and development were provided by the My500 jet printer.

5.2 Positioning System

The positioning System of the Demonstrator consists of three linear slides. Two of the slides are combined to construct a motion system for the camera movement in x- and y- direction. The third slide is used to move the PCB holder out of the examination area. This facilitates the change of test PCBs as well as adjustments at the camera like adaptions at the lens, since the camera is very close to the PCB (ultra-close range). The communication to the slides is established by an RS-232 interface and sending and polling ASCII commands and answers. Thus, the developed software sends commands to the slides and waits for the correct position answer prior to the image acquisition. The slides answer contains the position decoded in motor micro steps, which are transformed to micrometers for later adaption of the camera models for each acquisition position (camera's extrinsics).

5.3 Camera and Lens

For initial test runs a GigE camera was used. The library is still a part of the software, thus a GigE camera can be used if needed. The final system utilize a high speed camera (Baumer HXC40, Friedberg, Germany) with the Camera Link (CL) interface and the corresponding CL frame grabber from "Silicon Software" (Silicon Software GmbH, Mannheim, Germany). At the point of camera integration, this camera found to be the only one that satisfies the requirements. The camera is very compact ($52 \times 55 \times 52$ mm), light weight (180 g) and offers a high frame rate of 200 fps, if the right CL configuration is setup. The camera uses a 1 inch monochrome sensor with an isotropic pixel spacing of 5.5 μm and is equipped with a c-mount to attach lenses to the camera. Unfortunately, the camera comes without any Linux supporting software development kit.

Camera Link communicates via a serial port and ASCII commands similar to the RS-232 interface. Thus, a library for the camera communication was developed and integrated into the demonstrator software. At each software initialization the camera and frame grabber parameters are checked and set to the optimal values (e.g. shutter speed, gamma correction, CL tabs [Bau12]).
Corresponding to the examinations of the camera calibration a 10 mm lens is used in the final setup. The high precision lens comes from the manufacturer "Schneider Kreuznach" (Jos. Schneider Optische Werke GmbH, Bad Kreuznach, Germany) and is equipped with an threaded spacer. Thus, the distance from lens to sensor can easily be adjusted without removing the lens and placing single spacers.
The imaging setup yields a resolution of 40.3 line pairs per mm or 12.4 micrometers in x- and y-direction, which has been evaluated by imaging a USAF target.

5.4 Lighting

The first version of the demonstrator system uses a ring light around the lens to apply a homogeneous illumination of the PCB. Thus, an imaged solder joint gets an omnidirectional illumination. This results in a loss of structural surface information of the solder paste. For the texture based classification and feature detection the surface structure of the solder joints within the images needs to be enhanced. This is done by a lateral illumination. Two bar lights are used to illuminate the whole printing area (500 mm x 500 mm), to ensure a homogeneous lighting and to avoid a mounting of the lights onto the printer head, which would cause additional weight to be moved.

5.5 Demonstrator System

Figure 5.1 shows a CAD sketch and a photograph of the laboratory demonstrator system for the PCB analysis. Similar to the intended usage within the jet printer, the camera's (3) main axis is oriented orthogonal to the examination area or the calibration target, respectively. Two linear slides (5) simulate the x-y-motion of the print head and two lateral bar lights (1), which are controlled by the illumination module(4), illuminate the PCB (2). All parts are controlled by the inspection software (6).

5.6 Software Development

5.6.1 SolderInspect

The previous mentioned algorithms and hardware controls were encapsulated within a software with a graphical user interface. The final software is named *SolderInspect.* Since the jet printer runs on Linux, the software was developed on this operating system as well as using C++. The source code was managed within a Git-server for version control and contains the following modules, which can operate separately:

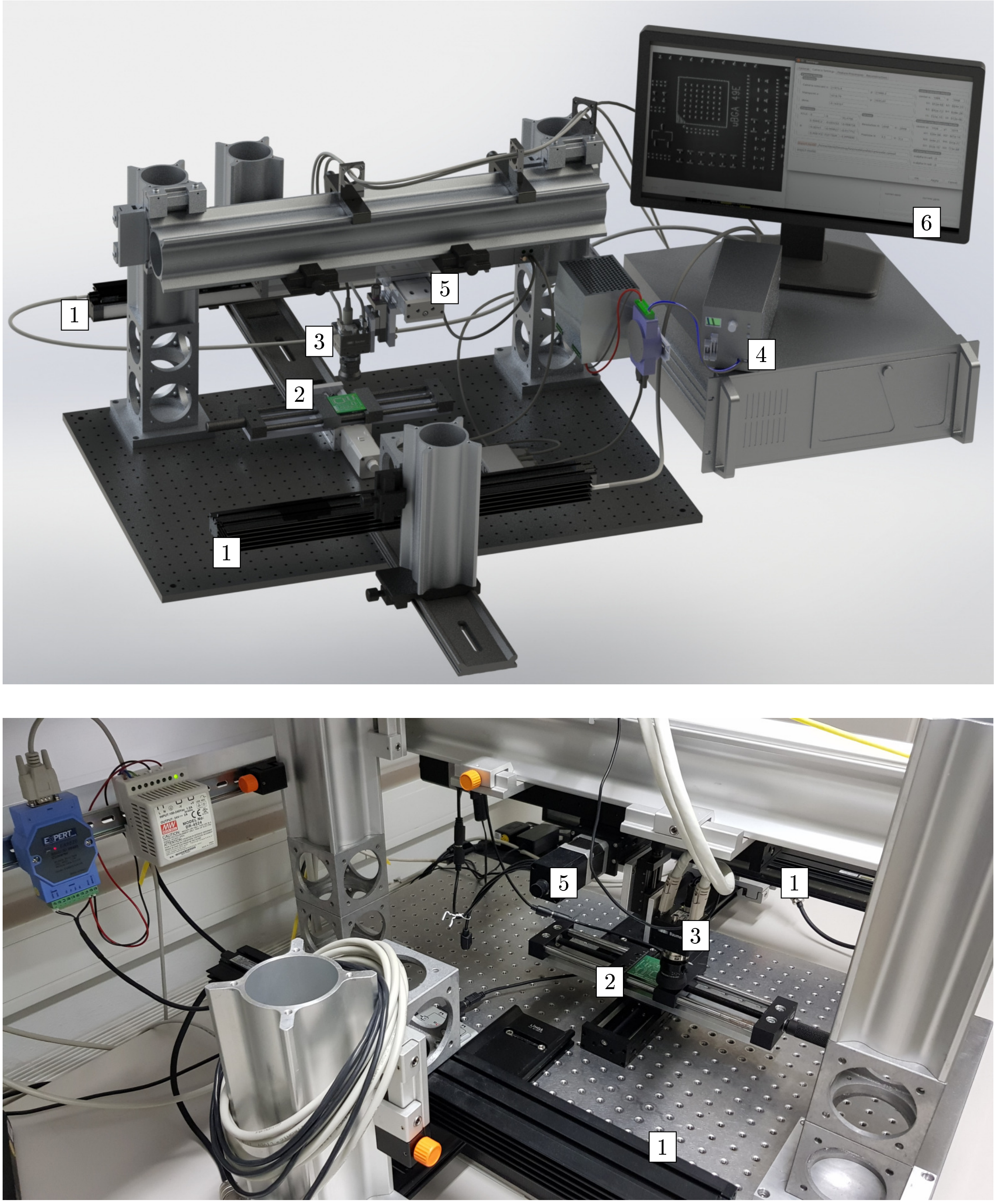


Figure 5.1 3D model (top) and photograph (bottom) of the demonstrator system: 1: bar lights, 2: PCB and PCB holder, 3: camera, 4: illumination control unit, 5: linear slides, 6: control and analysis software.

- graphical user interface
- camera control
- motion control (of the linear slides)
- illumination module

- camera calibration
- feature detection, matching and correction
- 3D reconstruction
- 3D post processing
- 3D visualization

The software was developed using the following programming languages and libraries:

- C++
- CUDA
- openCV
- openGL
- openMP
- Intel TBB
- Qt
- Eigen
- VTK (Visualization ToolKit)
- PCL (Point Cloud Library)
- Vimba (GigE Camera SDK from Allied Vision, Stadtroda, Germany)

SolderInspect consists of several main GUI windows. The first presented window is the image acquisition window from figure 5.2, that gives access to processing steps via buttons and a visualization of the current camera view. The second main windows is the 3D processing GUI. This includes the interaction with the 3D processing steps via drop-down menus, as well as the 3D visualization as shown in figure 5.3.

5.6.2 Processing Acceleration

Compute Unified Device Architecture (CUDA) NVIDIA's CUDA is a programming technique and language to parallelize algorithms extensively. Therefore, the data to be processed is transferred to the GPU, processed in parallel using thousands of processing units and is transferred back to the host. The functions that run on the GPU (device code) are also called kernel functions [JS10]. In this work CUDA is used at many stages of the processing pipeline. The highest advantage of this technique is given by openCV's GPU implementation of feature detection and matching,

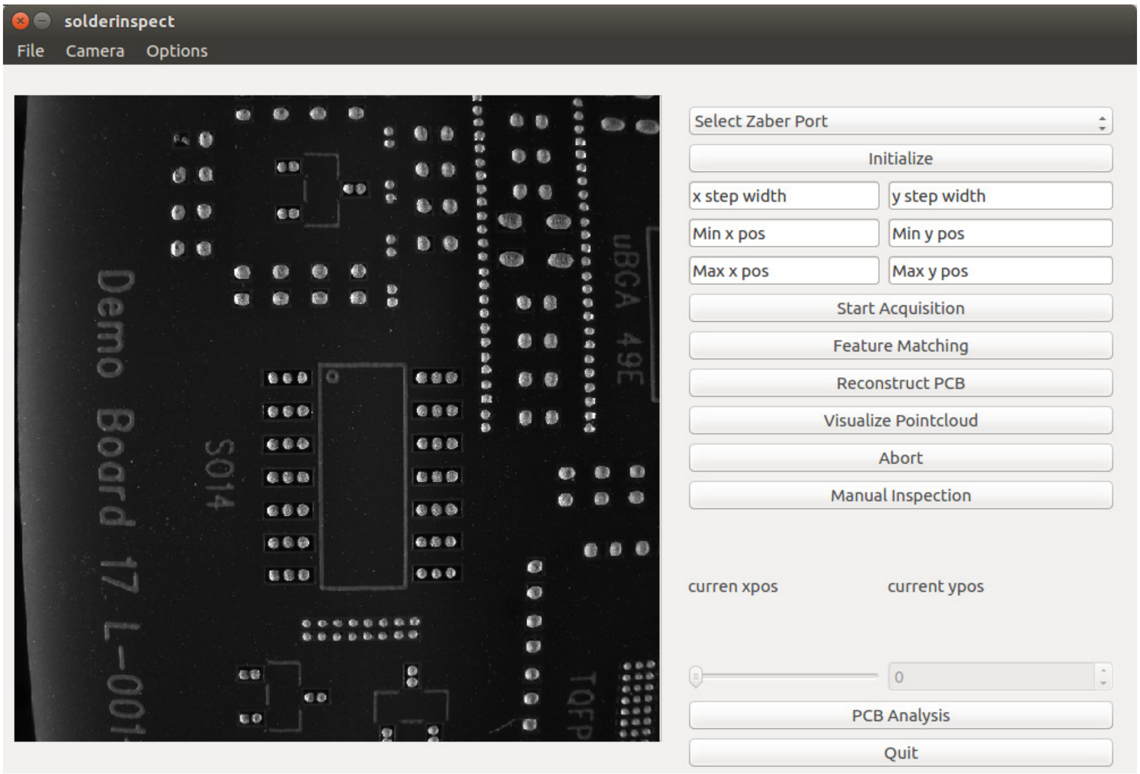
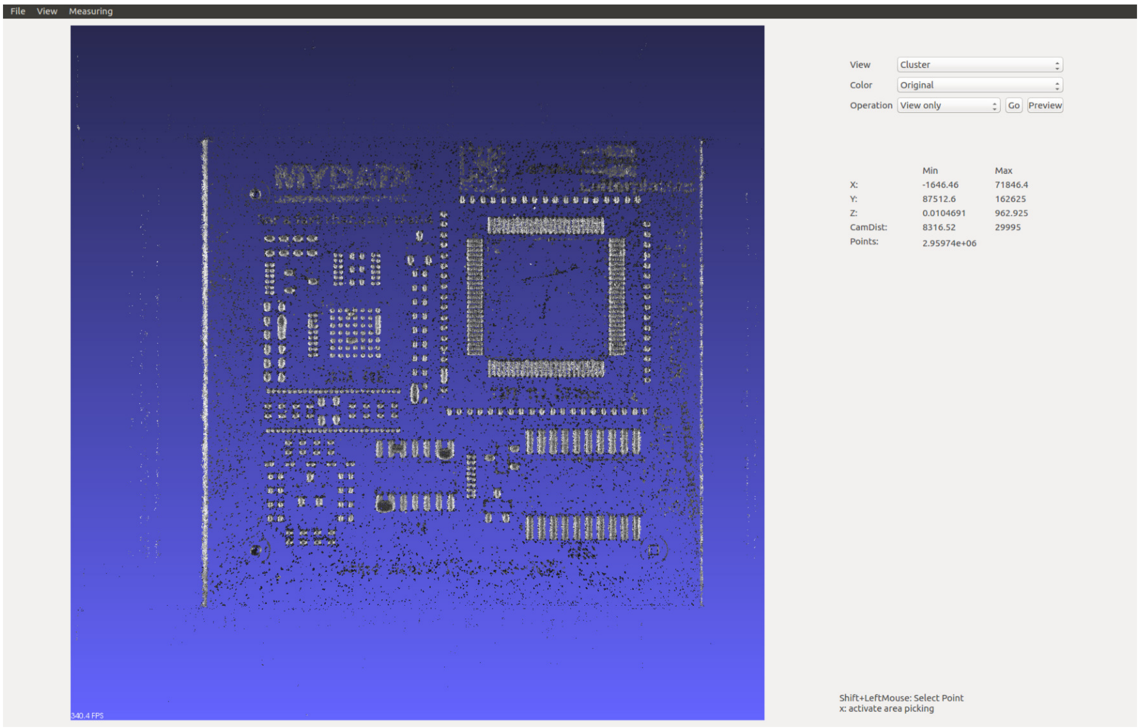


Figure 5.2 SolderInspect GUI: Main window



View: Cluster

Color: Original

Operation: View only

```
            Min            Max
X:          -1646.46       71846.4
Y:          87512.6        162625
Z:          0.0104691      962.925
CamDist:    8316.52        29995
Points:     2.95974e+06
```

Shift+LeftMouse: Select Point
x: activate area picking

Figure 5.3 SolderInspect GUI: Reconstruction window

resulting in a significant speed up. Besides these functions, own implementations of kernel functions (e.g. image projection onto the PCB plane) accelerate the processing significantly. The CUDA-based projection of one million data points of segmented solder joints from images onto the PCB-plane was done within less than 75 milliseconds.

Threading Building Blocks (TBB) - Pipeline Design Intels TBB is a programming technique that enables processing blocks to run in parallel within a specific pipeline design. Each separated part of this pipeline is called a TBB filter and has an input and an output interface. The TBB filter will be called only filter in the following explanations. Every filter runs on a single thread or if possible on it's own processor. This allows to start the image processing as soon as the first image is acquired or to start the image matching as soon as the first two overlapping images have been processed (e.g. in terms of feature detection). Not only the different filters can run in parallel, but also multiple instances of a filter [Rei07]. The image acquisition and the setup of an image queue for example is much faster than the subsequent feature detection. Therefore, the feature detection filter is allowed to run in parallel which results in a speed up of this step.

5.7 Online SPI Process

In this section the whole processing chain (Figure 5.4) within a program run is explained, starting from initializing over calibration and acquisition to the solder joint analysis. The developed software is capable to perform a step-wise user-triggered processing, automatic sequential processing or an automatic online processing. Since the last option is the one with the best performance and most relevant setting for the intended usage within the PCB jet printer, the following descriptions are referred to the online settings of the software.

Setting up the System At the start up of the system, default parameters are loaded. These come from the examinations of reconstruction, feature detection, matching and correction, camera calibration and processing acceleration. Nevertheless, the software contains a settings GUI to change all parameters of the system. These are the PCB dimensions, processing modes, images per object point, image saving options (bin, jpg, png, bmp, appending camera model to image data set), camera modifications (every parameter like focal length, rotation matrix, etc. can be adapted, import and export of camera models, setting camera mode - free run and triggered) and feature processing (detector selection, parametrization, descriptor selection, matching selection, correction method). For a better user-friendliness these settings are distributed under the settings tabs *general*, *camera*, *features* and *reconstruction*. The corresponding user interfaces can be seen in the appendix in figure A.13, A.14, A.15 and A.16 at pages j to l. At least the PCB dimensions are needed to compute the scanning region and image density using the camera model. During the initialization the software detects automatically, if a GigE or CL camera is in use and sets its parameters, afterwards the linear slides are sent to the home position for coordinate system establishment. The software is highly user informative and gives visual outputs above the cur-

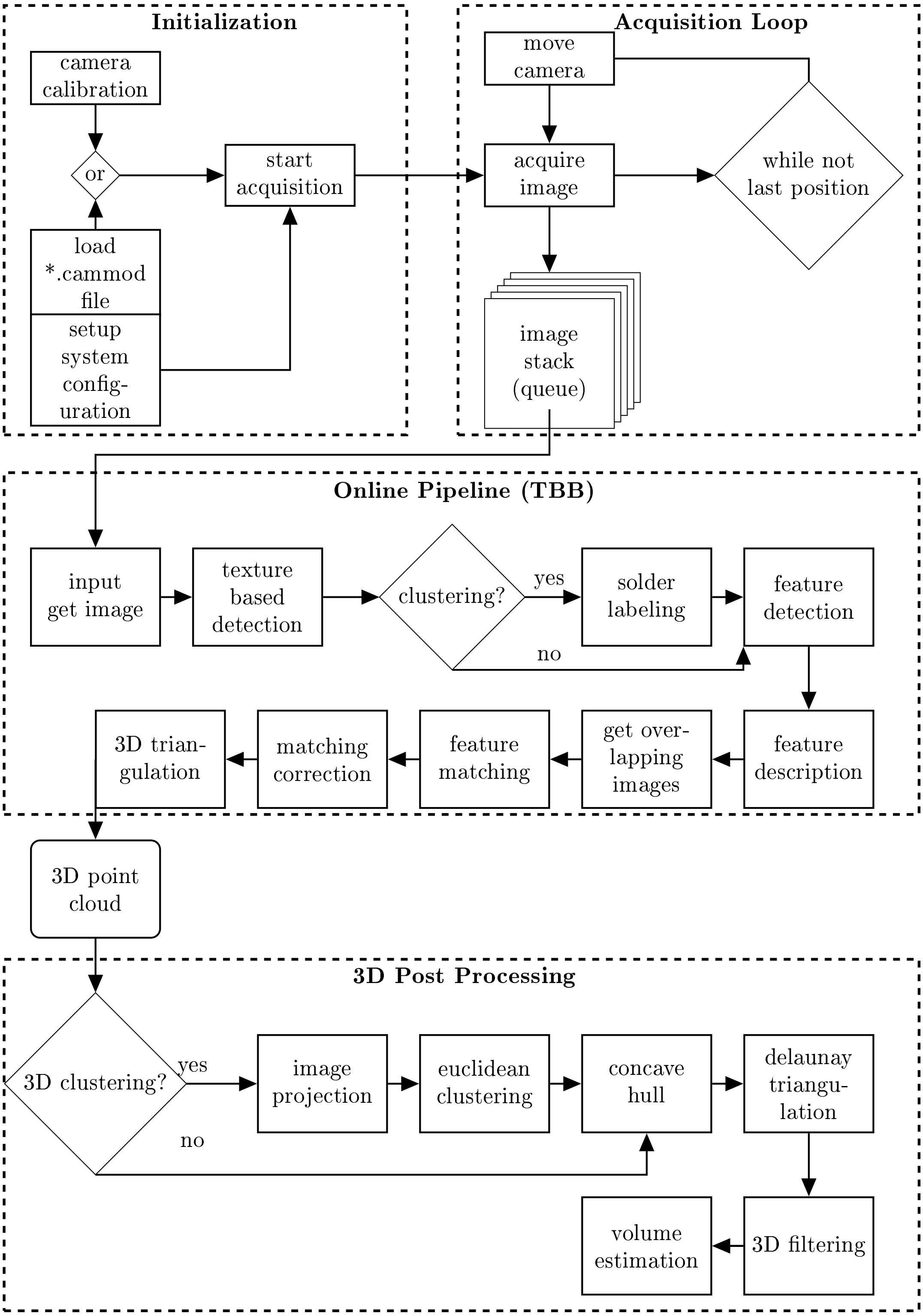


Figure 5.4 Flow Chart of the Online SPI Process

rent processing. In the appendix the user interface during camera detection and initialization is presented in figure A.17 and A.18.

Camera Model Setup If no camera model was computed using the software camera calibration tool, the user has to run the camera calibration prior to the inspection process. Once a model is computed, it can be saved by the user within the camera drop down menu, the camera settings tab or when saving the data set if the option "save camera model with data set" is activated in the general settings tab. If the calibration tool is started, the software prompts the user to place the 3D calibration target at the PCB holder. After a confirmation by the user the camera moves to the demonstrators center position and acquires an image of the calibration target. Afterwards the detection runs and estimates the magnification using the detected rectangle of the target. Dependent on this magnification, 8 imaging positions are calculated where the camera moves to and acquires calibration images. The multi-image calibration starts automatically after the last image is sent to the software. Afterwards the camera model is ready for feature correction and reconstruction.

Processing Modes Three different modes are available within the software: manual, semi-automatic and online.
Manual: The software is equipped with an manual mode for examination reasons. Within these settings the software performs only one operation on user request. This includes an additional camera control interface (see figure A.19), where the user can specify if the camera runs in a triggered mode, where a software trigger is used to instruct the camera to acquire an image or in free run mode, where subsequently images are captured and presented to the user according to the frame rate which depends on the connection settings and shutter speed. The shutter speed, gamma correction and illumination brightness (0 to 255) can also be set by the user in this control interface, as well as the movement of the camera to arbitrary positions and to the maintenance position, where the camera moves towards the user and the PCB holder is moved away from the camera to allow for maintenance at the camera without its disassembly. All other processing parts can also be performed detached from each other. However, some steps require other parts to be performed first, e.g. reconstruction is not available without feature detection and matching. Therefore, the software checks if necessary previous processing steps were done and informs the user if additional steps are necessary.
Semi-Automatic - Sequential: Within this mode, the user sets all parameters for the processing pipeline as proposed. Then the software performs all parts subsequently and stops after every step. Thus, at first the image acquisition (or data set loading) is performed completely and the user is informed afterwards. After a confirmation the feature detection takes place and so on. Thus, the pipeline can be examined after each single step since no parallel processing is done.
Online: The most relevant processing mode is the online mode of the system. In that mode, the image acquisition and all essential steps of the processing pipeline are performed in parallel within the TBB online pipeline. This will be described in further detail in the following paragraph.

Inspection If the camera model and settings are set up the software calculates the scanning region and the step width in x- and y- direction. The values are presented to the user in main acquisition window and a click on "start acquisition" initiates the inspection process. In online processing mode the 3D visualization window comes up, which displays the online reconstructed point cloud. The software moves the camera to the first acquisition position and starts the scanning process. After every image acquisition the camera model is modified considering the current camera position (obtained from the linear slides) and the position is stored to a custom "*.posdat" file. The newly acquired image is pushed to a data queue waiting for further processing. This data queue is the input interface to the online processing pipeline. As soon as the pipeline detects a new image in the queue and if there is free processing capacity (depending on the pipeline configuration) the image is pulled from the queue and a thread for a new processing pipeline is started. Each pipeline thread holds its on data package to handle data that is needed across multiple filters and is pushed from one filter to the subsequent one. The first filter takes the camera models distortion parameters and corrects the images. If the image based clustering is selected, the filter for detecting solder joints also assigns unique integer labels to the solder joint regions. This filter waits until the previous image has been processed in solder paste detection and labeling to avoid the double assignments of labels. In practice no waiting time is observed since the GPU based detection and labeling runs faster than the image acquisition. The region centroid is projected into the previous image and the assigned label is used, if the solder joint was already acquired, to obtain a unique label for a solder joint across multiple images. If the joint is acquired for the first time, the label counter is increased and assigned to the solder joint area. After labeling the image, it is inserted into the data package and the feature detection filter is executed. If the labeling was done in the previous filter each detected feature also gets the numeric label from its corresponding solder joint. Subsequently, the features are appended to the data package and pushed to the feature description filter. The next filter takes the n-th image and computes for all $n-1$ images if there is an overlap of the images, since only overlapping images can be used for the photogrammetric reconstruction and the overlap indices are stored in the pipeline data package. Using these overlap image indices the matching filter is executed only on images that have PCB regions in common. The obtained feature matches are pushed to the correction filter were the camera model based elimination of false matches is performed. Finally, the last TBB-filter performs the model based triangulation using the corrected feature matches and the adapted camera models. The result of the parallel processing chain is the 3D point cloud representing the reconstructed solder joints. Afterwards, the 3D post processing is performed. However, these steps are not part of the parallel processing chain, since the reconstruction of the point cloud must be finished prior to any post processing steps as clustering (if no labeled points are used), vertex point processing, filtering and volume estimation. These steps are performed consecutively after the parallel pipeline is finished. Due to GPU acceleration, this is the case shortly after the last image has been acquired.

File Types Two essential file types were developed during the examination to facilitate the software usage and to improve the work flow of the examinations. The first file type is the "*.posdat". This file contains the position data of the x- and y- motor slide and is saved in combination with the acquired image data set. This ensures a re-usability of the data set, for example for a repeated processing under different parameter settings without a repetition of the image acquisition process.

The second file is the "*.camod" file which contains all camera parameters as follows:

- x- and y- position of principal point
- principal distance in x- and y- direction
- shear factor
- x,y,z coordinate of camera center
- 9 rotation matrix entries (order by rows)
- x,y- coordinate of lens distortion center
- 4 radial distortion coefficients
- 2 tangential distortion coefficients
- x- ,y- coordinate of principal point (for inverse distortion)
- 4 inverse radial distortion coefficients
- 2 inverse tangential distortion coefficients
- sensor pixel size in x and y-direction

Due to this file, the calibrated camera model is available at any time without a new camera calibration prior to each acquisition and reconstruction.

Properties The final system can be characterized by the following parameters:

- acquisition:
  - sensor pixel spacing: 5.5 μm
  - sensor type: monochrome CMOS, global shutter
  - fps: 200
  - 10 mm focal length lens
  - f/number 1.9 - 16 (16 is used for maximum depth of field)
- reconstruction:

  - resolution: 12.4 μm (x), 12.4 μm (y), 25 μm (z)
  - area: variable (500 x 500 x 1 mm - x,y,z);

6 Results

6.1 Camera Calibration

6.1.1 Image Pre-Processing

The results of the pre-processing steps can be seen in figure 6.1. Image A is the input image, which is given to a shading correction. In image B small reflections have been removed. After the scale space filtering (C and D) all artifacts are removed and the markers are obtained by the Otsu threshold in image E. The filtering of image properties only fully visible markers remain.

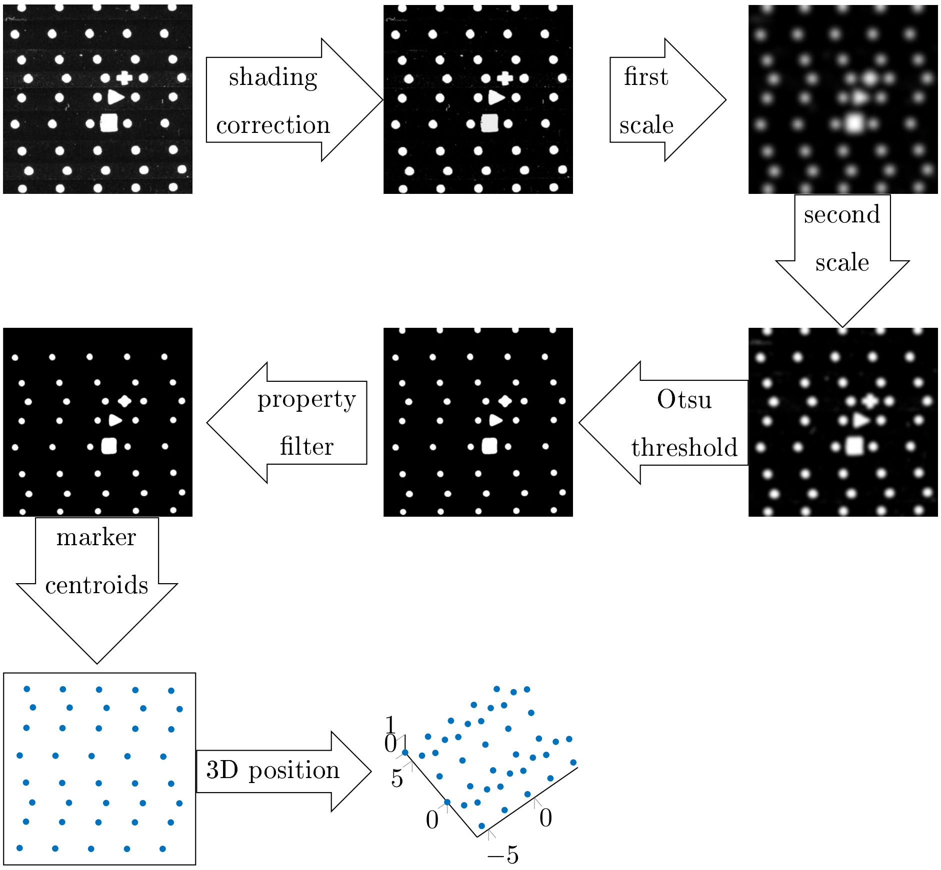


Figure 6.1 Example of the image pre-processing pipeline of the calibration target.A: input image; B: result of the shading correction; C: first scale space; D: second scale space; E: Otsu thresholding of the last scale space, F: resulting markers after the filtering of image properties, G: coordinates of detected markers, H: 3D assignment of the markers.

6.1.2 Initial Geometry based Distortion Correction

In the presented assignment of image markers to world coordinates the markers are extracted on lines. Lens distortion complicates this assignment. After the proposed correction, a rectified image is obtained. Figure 6.2 shows the decrease of the mean value deviation of all angles α_{ij} between the vectors v_1 and v_2 from 90°. The result of the image rectification by correcting the image points is visible in figure 6.3. The developed method for an initial geometry based distortion correction

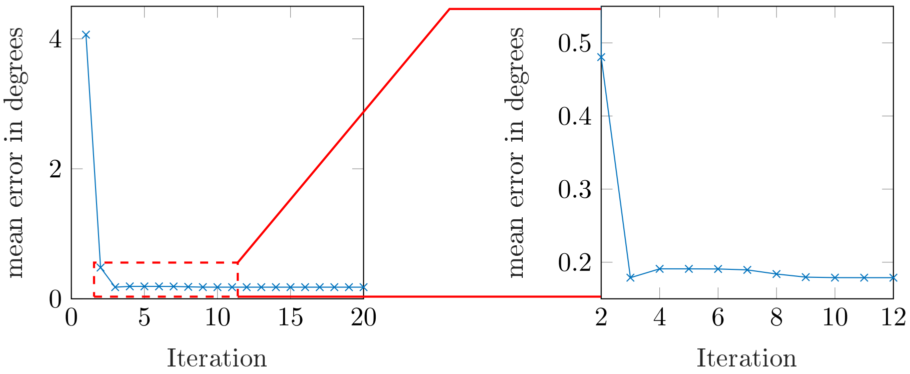


Figure 6.2 Decrease of angle error ($\alpha - 90°$) within the 3D target; left: all iterations, right: iteration 2 - 12.

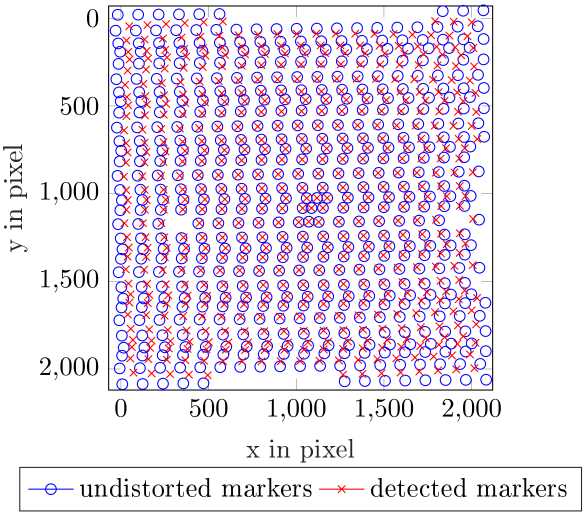


Figure 6.3 Distortion correction of 3D target; circles: detected markers, crosses: corrected (undistorted) markers.

could remove the lens distortion effects from the calibration image. After 5 iterations this method converges to a minimum error of 0.174 degrees while the distorted image contains a mean angle of 94.5 degree for a 10 mm lens, which results in the most significant distortion. After 12 iterations no change in the distortion coefficients and thus in the error was observed. The corrected markers lay on straight lines along the x coordinate, whereas the originally detected markers establish a curve. This correction results in a proper assignment of point correspondences for the proposed calibration methods and can be adopted to other known calibration target geometries. The effect of the correction can be seen in figure 6.3. The detected marker positions (red crosses) move from a curve to straight horizontal lines (blue circles). This effect is more visible at the edge of the image, where the distortion is more present.

6.1.3 Comparison of Reprojection and Reconstruction Error

All calibration methods were evaluated by the reprojection error and the reconstruction error. Therefore, the known 3D points were projected into the image and the 2D reprojection error between the detected and projected point is calculated using equations 3.23 and 3.24 (see figures 6.4 and 6.5). In the second evaluation, the obtained models are applied to reconstruct the designed 3D target (calibration markers) and calculated the total 3D reconstruction error, as well as the error along the x-y- plane and along the z direction. The results of the best models and imaging setup (lens and spacer) are represented in table 6.1 for the total reconstruction error ϵ_{recon}. Table 6.2 splits these results into x-y and z error for the same configurations. It is evident from figures 6.4 and 6.5, that the small reprojection error ϵ_{repro_s} (e.g. within the single image 2D calibration) does not correspond to a good camera model in terms of reconstruction capability. The corresponding model has a reprojection error ϵ_{repro_s} below 0.1 pixel and a reconstruction error ϵ_{recon} of above 1 millimeter, while the multi-image calibration using the 3D target delivers a reprojection error ϵ_{repro_m} of 1.1 pixels (10 mm lens, 5.97 spacer) with a reconstruction error ϵ_{recon} of 14 micrometers.

Table 6.1 Best 3D reconstruction error ϵ_{recon} in μm and used spacer in mm.

Lens	10 mm		16 mm		25 mm	
Method	ϵ_{recon}	spacer	ϵ_{recon}	spacer	ϵ_{recon}	spacer
MCCo	2.3	5.9	15.3	10.1	15.0	16.2
MC	14.0	5.9	15.5	10.1	20.6	22.3
SCCo	183.0	5.9	363.1	10.1	128.1	10.1
SC	203.0	3.9	416.6	10.1	765.5	10.1
SC2D	1172.3	4.4	1260.5	10.1	1580.5	10.1
Zhang	221.2	5.9	1398.1	10.1	6477.1	10.1

Table 6.2 Reconstruction error ϵ_{recon} in μm for (x,y) and z directions.

Lens	10 mm		16 mm		25 mm	
Method	(x,y)	z	(x,y)	z	(x,y)	z
MCCo	1.4	1.8	6.0	13.5	4.2	13.9
MC	6.4	11.9	6.4	13.2	2.2	20.4
SCCo	176	45	275.4	236.4	640.3	29.0
SC	184.0	85	285.5	303.2	688.6	334.1
SC2D	406.3	1098.9	583.5	1116.5	1163.6	1067.3
Zhang	209.7	57.2	1396.2	56.4	6451.0	459.2

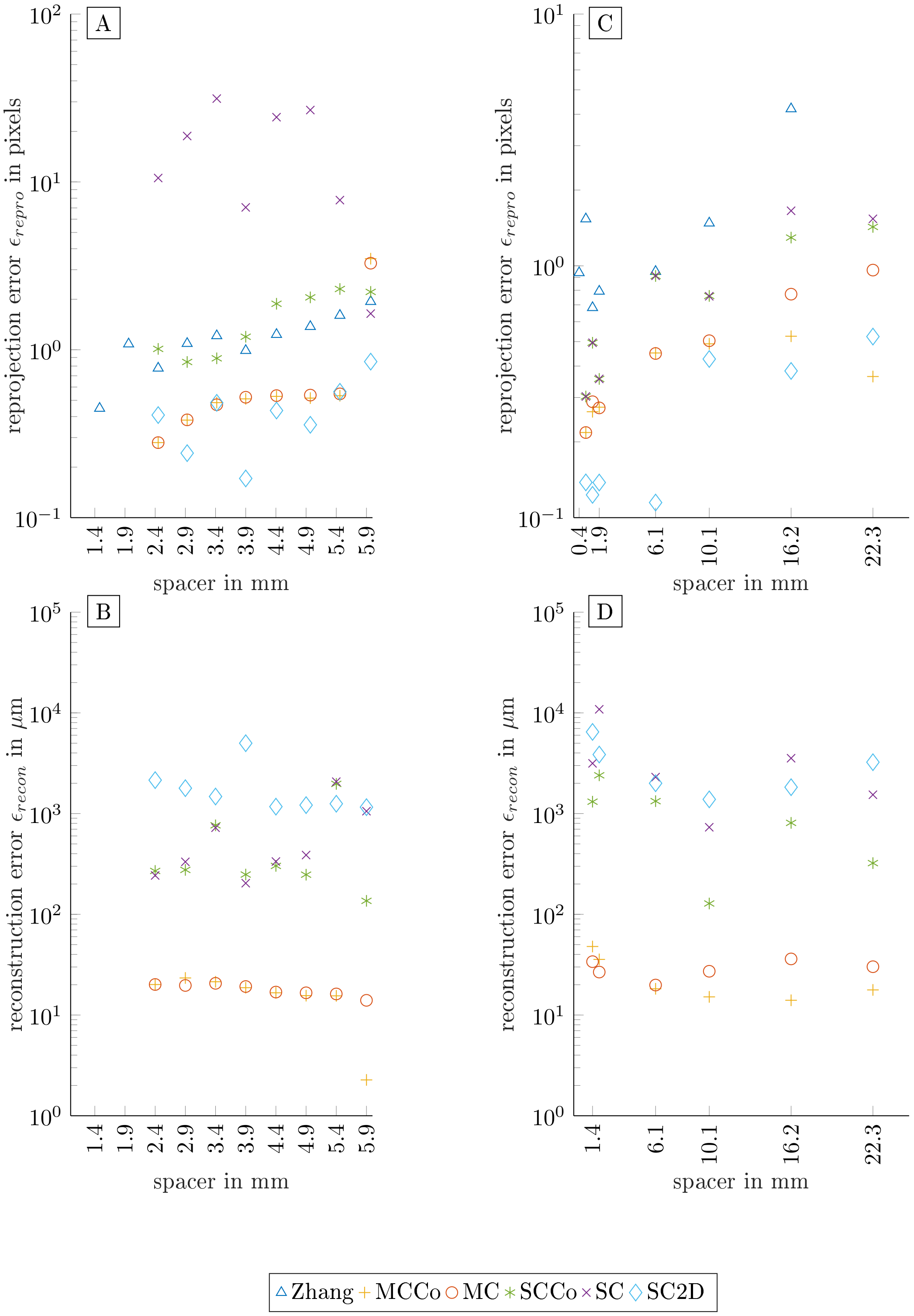


Figure 6.4 Error of reprojection and reconstruction of the 10 mm and 25 Lens: A: reprojection error ϵ_{repro}; B: reconstruction error ϵ_{recon} of the 10 mm lens; C: Reprojection error ϵ_{repro}; D: reconstruction error ϵ_{recon} of the 25 mm lens.

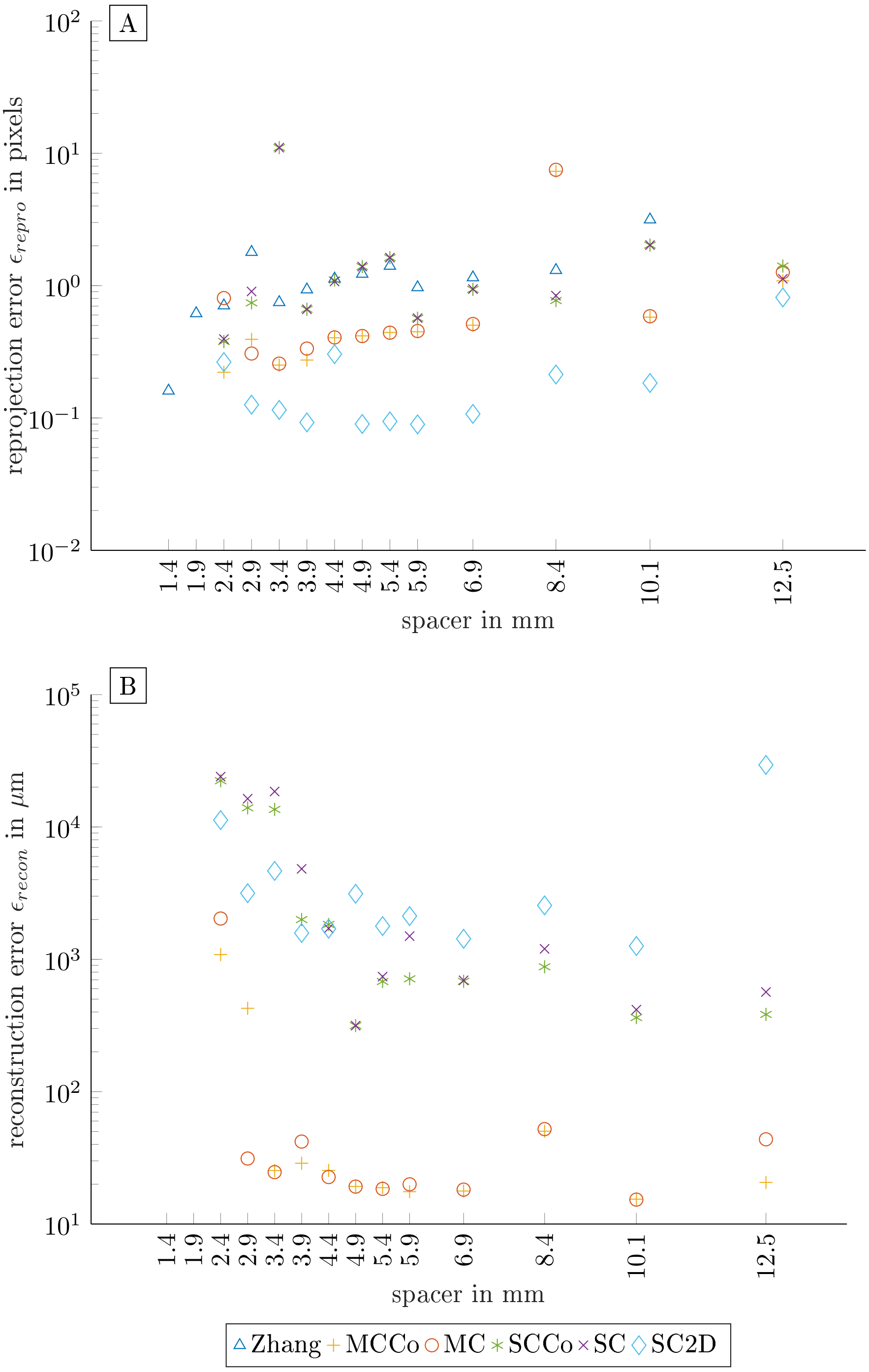


Figure 6.5 Error of reprojection and reconstruction of the 16 mm Lens: A: Reprojection error ϵ_{repro} and B: reconstruction error ϵ_{recon} of the 16 mm lens.

As shown in tables 6.1 and 6.2, a reconstruction error of 2.3 μm (1.4 μm lateral and 1.8 μm transversal) with the 10 mm lens and the largest (5.9 mm) spacer was achieved when using the multi image calibration with iterative marker correction (MCCo). This result was even much better than the reconstruction from Zhang's calibration with an reconstruction error of 221 μm. The methods MC (multi image calibration), MCCo (multi image calibration with iterative correction), SC (single image calibration), SCCo (single image calibration with iterative correction) use the 3D target, whereas SC2D performs a single image calibration using the 2D circular pattern target and a orientation varied chessboard pattern is used by Zhang's calibration method (openCV implementation).

6.1.4 Effect of Distortion Parameters

Figure 6.6 shows the reconstruction as well as the reprojection error of the 10 mm lens setup using up to four radial distortion coefficients and two additional decentering coefficients, excluding the errors where no lens model is used, since those are significantly higher. The reprojection error as well as the reconstruction error show a higher value when using only one distortion coefficient (k1). There is no significant change in the errors if more than two coefficients and additional tangential distortion coefficients are used. Considering at least two distortion coefficients, the use of spacers leads to an increase of of the reprojection (except for the 4.4 mm spacer) error but in turn to a decreased reconstruction error, where the smallest error is achieved by using the largest spacer (5.9 mm). The number of distortion parameters has a restricted influence on the resulting camera

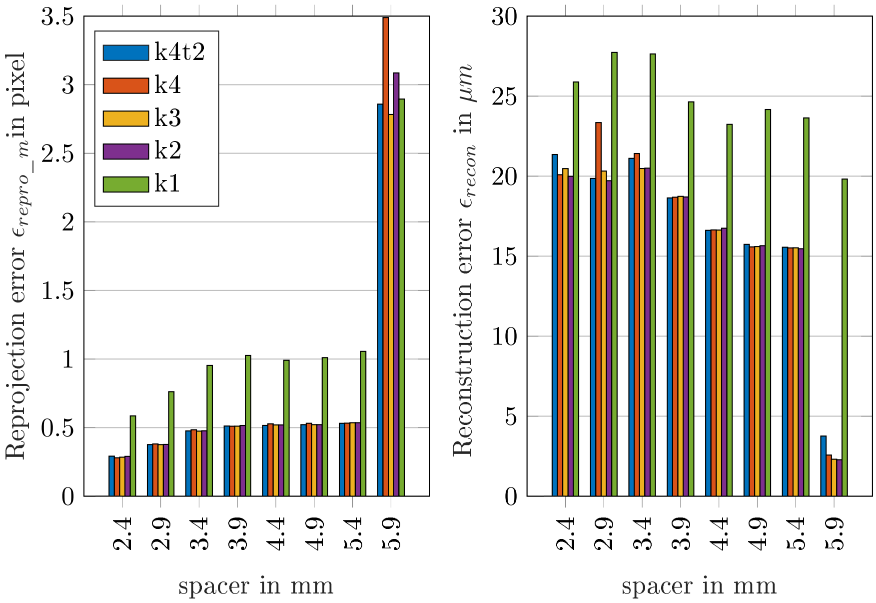


Figure 6.6 Reprojection error ϵ_{repro_m} (left) and reconstruction error ϵ_{recon} (right) for the 10 mm lens, k1 - k4: use of 1 - 4 radial distortion coefficients; k4t2: 4 radial and 2 tangential distortion coefficients are used.

model. Owing to these results, two parameters are considered as sufficient.

6.2 Texture based Solder Joint Detection

6.2.1 Pre-Processing

An example of the input image and the resulting image of the gradient computation is shown in figure 6.7. After the gradient computation the image is thresholded using the Otsu's method. An exemplary result can be seen in figure 6.8.

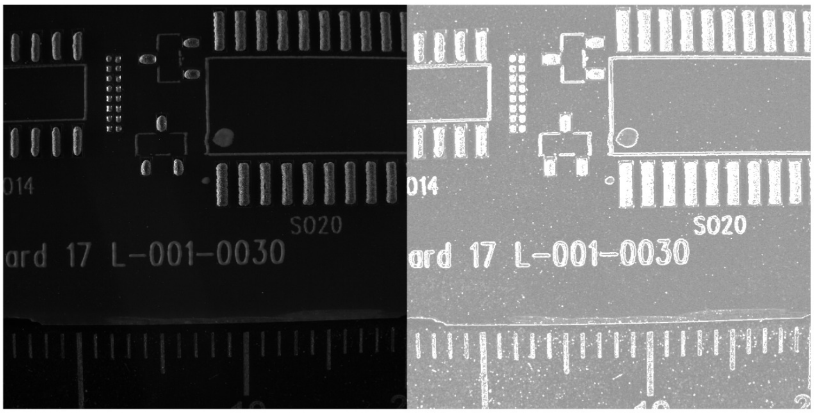


Figure 6.7 Exemplary result after gradient computation: original image (left) and gradient image (right).

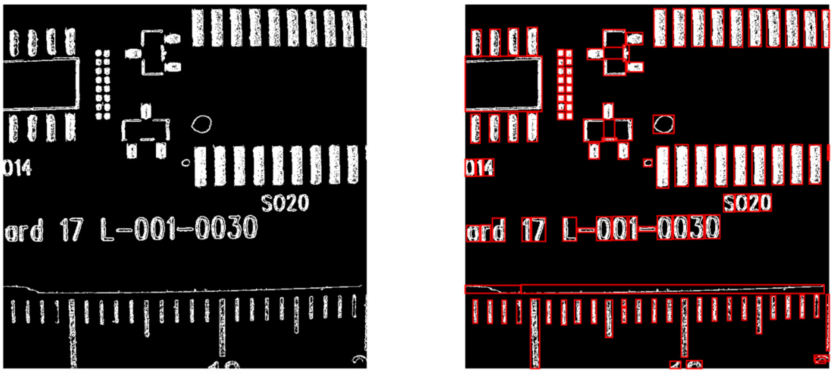


Figure 6.8 Exemplary result after thresholding the gradient image (left) and corresponding bounding boxes (right).

6.2.2 Classification

From the obtained regions all 14 Haralick features were calculated to examine each feature if a separation of solder joint and non solder joint regions is possible. 5 of 14 parameters were found to allow for this separation, which leads to a proper classification of regions into solder and non-solder.

The suitable parameters are *contrast, variance, homogeneity, energy* and *entropy*. The results of the 5 parameters in 14 randomly selected solder joint and non solder joint regions are shown in figures 6.9 for the contrast, 6.10 for variance, 6.11 for the energy parameter, 6.12 for homogeneity and figure 6.13 shows the result of the entropy parameter.

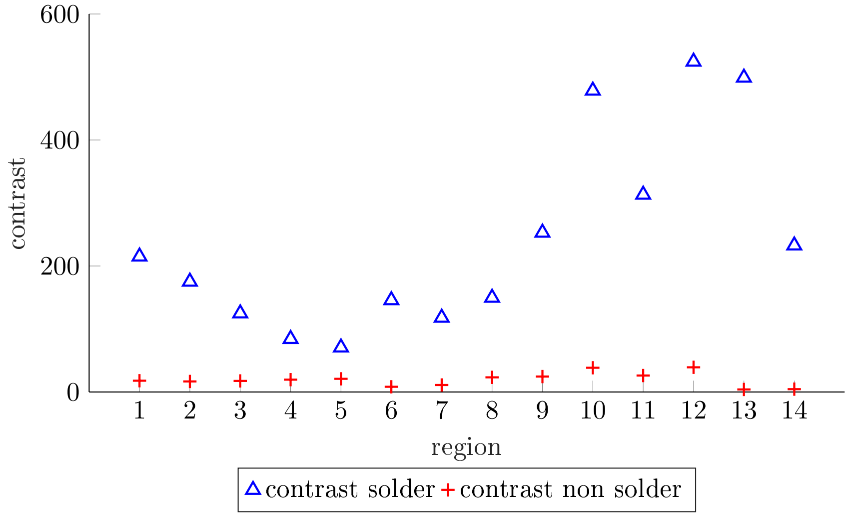


Figure 6.9 Comparison of contrast values of solder paste (blue) and non-solder structures (red). Regions were randomly selected. The contrast of solder joints is above the contrast of non-solder structures in any case, which allos for a unique separation.

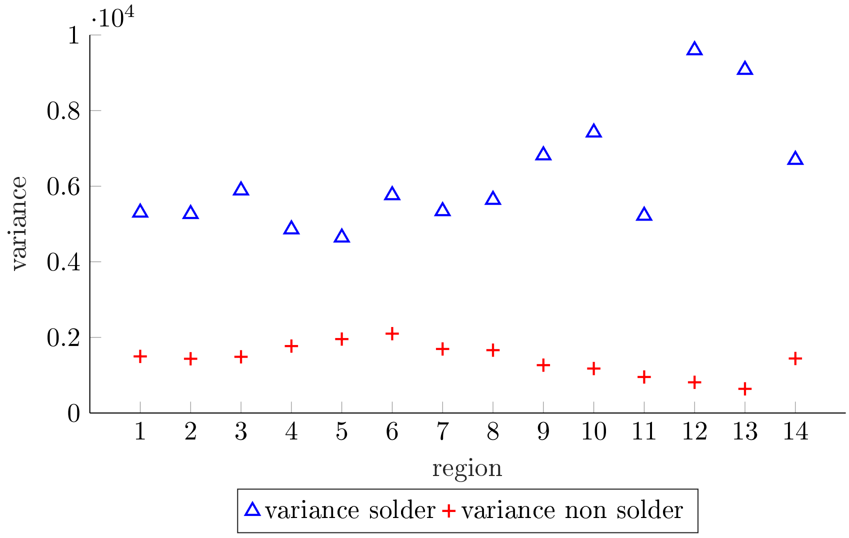


Figure 6.10 Comparison of the Haralick feature variance of solder paste (blue) and non-solder structures (red). Regions were randomly selected. The variance of solder joints is above the variance of non-solder structures in any case, which allows for a unique separation.

After examining these Haralick parameters, only regions containing solder joints do remain in the result image, as it can be seen in figure 6.15. The remaining parameters were not suitable for a classification since the resulting values are not separable. An example of a Haralick feature

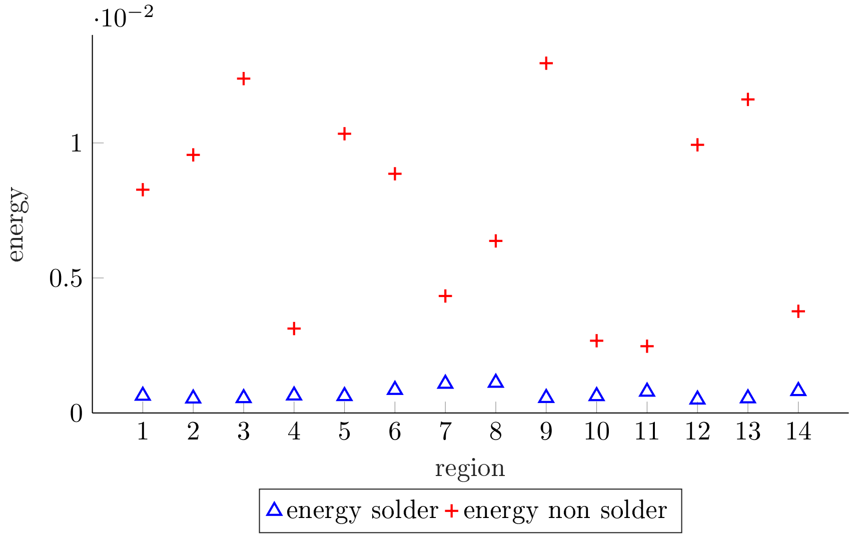


Figure 6.11 Comparison of energy values of solder paste (blue) and non-solder structures (red). Regions were randomly selected. The energy of solder joints is above the energy of non-solder structures in any case, which allows for a unique separation.

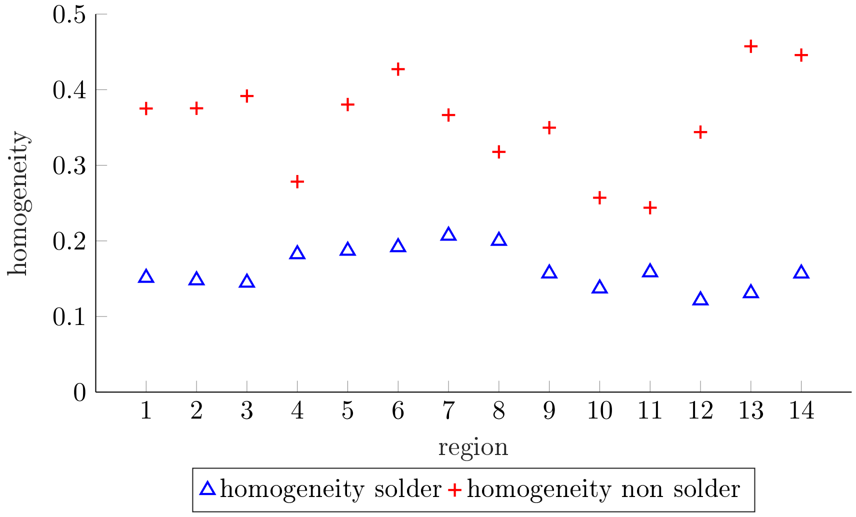


Figure 6.12 Comparison of homogeneity values of solder paste (blue) and non-solder structures (red). Regions were randomly selected. The homogeneity of solder joints is above the homogeneity of non-solder structures in any case, which allows for a unique separation.

(correlation) that does not work for classification is given in figure 6.14. The result of all remaining Haralick parameters are shown in the appendix on page e in figures A.4 to A.11.

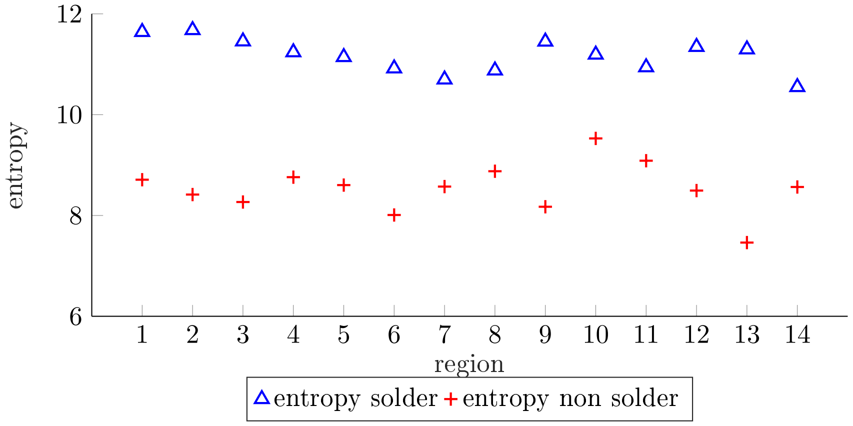


Figure 6.13 Comparison of entropy values of solder paste (blue) and non-solder structures (red). Regions were randomly selected. The entropy of solder joints is above the entropy of non-solder structures in any case, which allows for a unique separation.

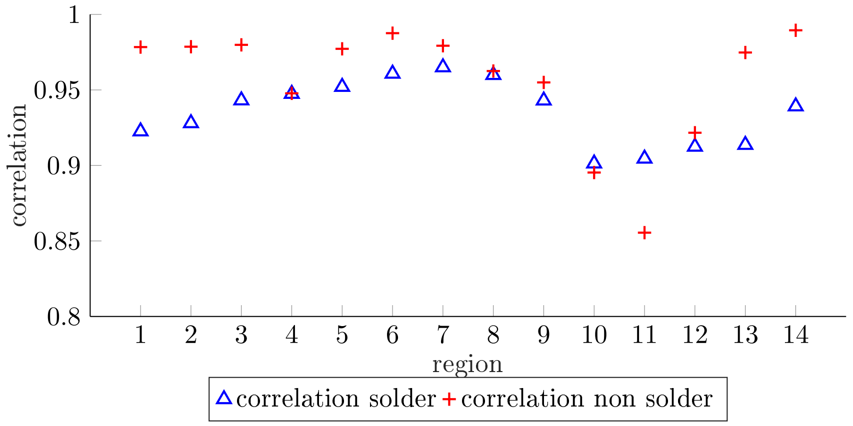


Figure 6.14 Comparison of correlation values of solder paste (blue) and non-solder structures (red). Regions were randomly selected. The correlation of solder joints is mixed with the correlation of the segmented non-solder areas. Thus, a separation of the two classes is not possible.

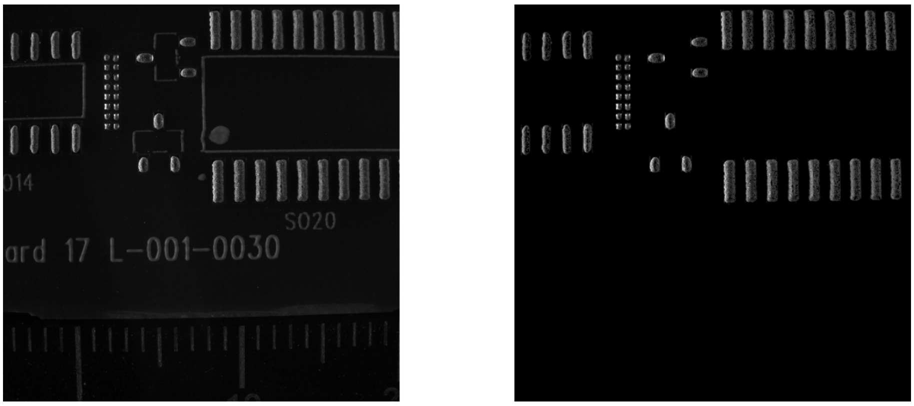


Figure 6.15 Classification result of the example image: original image (left) and the image with remaining classified solder paste regions (right).

6.2.3 Detection rate

A total number of 27335 solder joints was examined with the proposed classification method. Taking every single image as a stand alone image a detection rate as shown in table 6.3 is established. The three data sets result from the different PCB and illumination conditions as presented in section 4.1.1. Data set 1 contains a one-sided illumination of the PCB with a lot of dust particles. The second data set is obtained from a clean PCB and a one-sided illumination, whereas the third data set shows a clean PCB surface that is imaged with a two-sided illumination. Due to the photogrammetric normal-case setup and scanning principle each solder joint is captured by at least two camera positions. Taking these multiple images into account, by considering a solder joint to be detected if it is detected in at least one image, and re-evaluating the detection rate, an improved result is obtained. The corresponding values are shown in table 6.4. The detection rate for each data set can be seen in the box plots in figure 6.16.

Table 6.3 Overview detection rate

	data set 1	data set 2	data set 3
total solder joints	11664	7790	7881
true positives (TP)	11459	7367	7772
false negatives (FN)	205	423	109
false positives (FP)	265	6	31
sensitivity [%]	98,24	94,57	98,62
PPV [%]	97,775	99,92	99,6

Table 6.4 detection rate with multiple images

	data set 1	data set 2	data set 3
total solder joints	11664	7790	7881
true positives (TP)	11641	7790	7880
false negatives (FN)	23	0	1
false positives (FP)	265	6	31
sensitivity [%]	99.80	100	99,99
PPV [%]	97,779	99,92	99,61

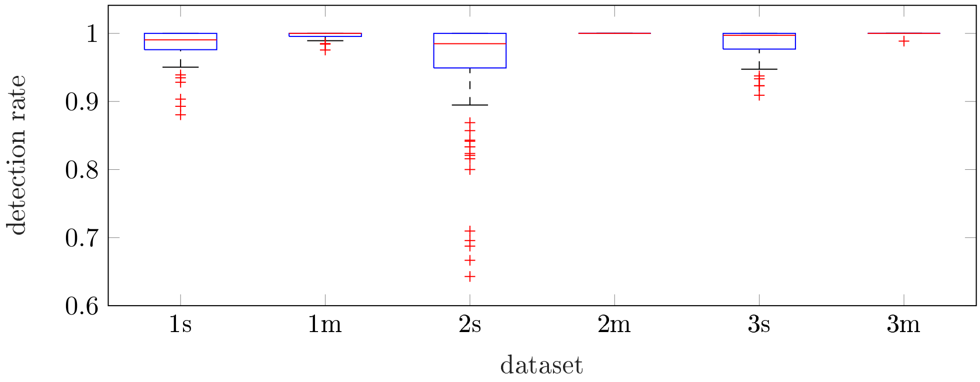


Figure 6.16 Box plot of detection rate in 3 data sets - 1: single sided illumination, large amount of dust on PCB; 2: single sided illumination, clean surface; 3: double sided illumination, clean surface. s indicates the evaluation of the detection for every single images, m indicates the usage of multiple images.

6.3 Feature Detection

6.3.1 Correlation of Camera Integration Time and FAST Feature Detection

By examining the introduced SolFeC parameter (equation 4.28), a proper camera integration time of 13 ms was determined. The behavior of the SolFeC-parameter is given in figure 6.17 A. An example of feature coverage and the obtained parameters is visible in the appendix in figure A.20. Figure 6.17 B shows the independence of the developed SolFeC parameter from the solder joint size (see examples in figure 4.5). The number of total keypoints detected in the whole image becomes maximum at an integration time of 19 ms, whereas the ratio of solder joint and background keypoints shows the maximum at 13 ms (figure 6.17 C). Independent from these keypoints, the number of unusable background keypoints increases constantly with the integration time. This is due to the higher contrast of dust particles, the slightly structured surface and scratches. In turn the solder joint structure decreases due to saturation resulting in a smaller amount of solder joint keypoints with higher integration time above 13 ms. These facts lead to an overall decrease of the ratio between solder joint and non solder joint (background) keypoints. Computation time is clearly correlated to the total number of detected keypoints, since more keypoint candidates have to be examined, whereas low integration times lead to low contrast and non-keypoint candidates are removed in the early detection stage (figure 6.17 D). In the remainder of the thesis an integration time of 13 ms is used.

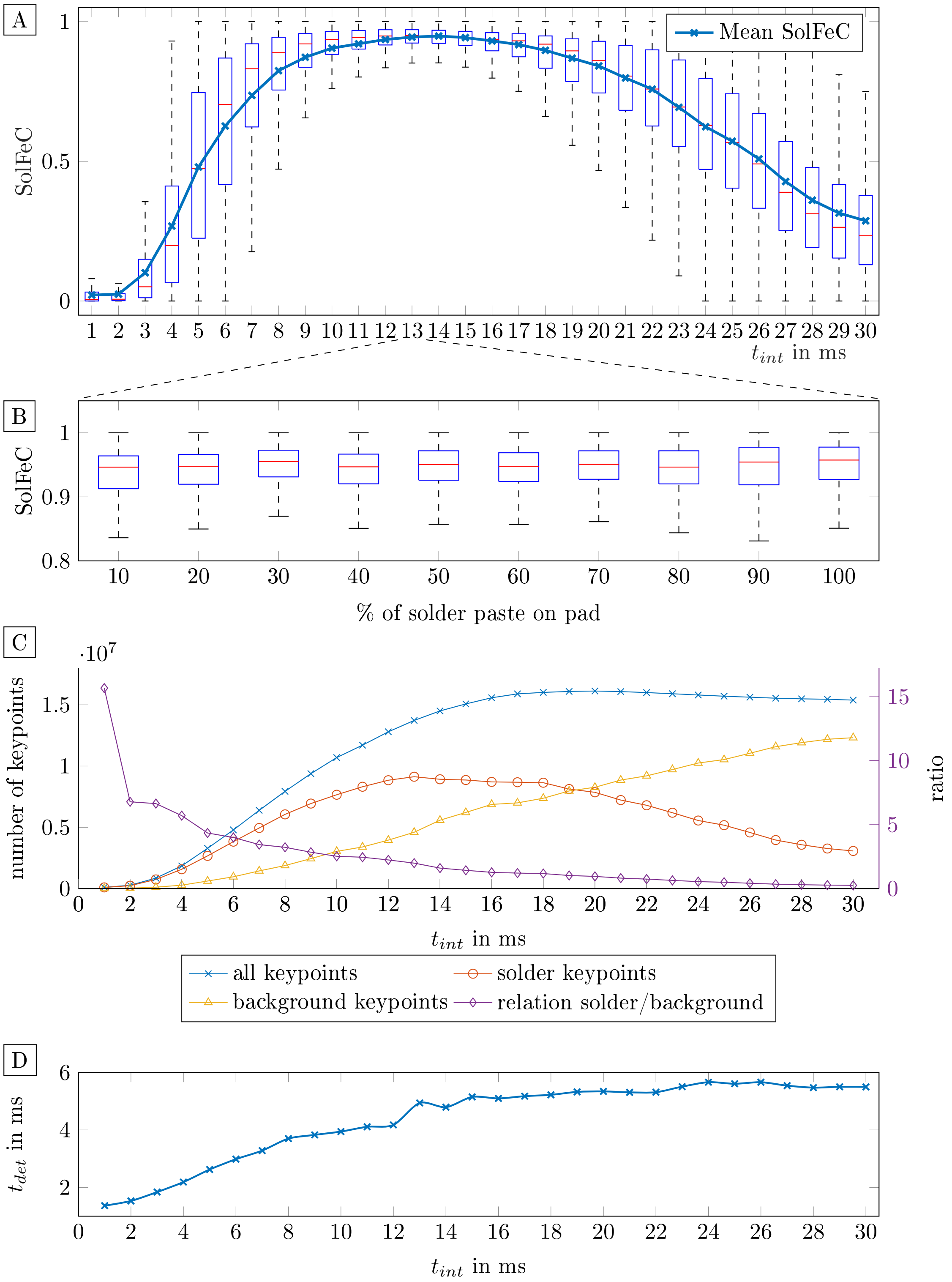


Figure 6.17 A: Correspondence of camera integration time and SolFeC for FAST feature detection; B: SolFeC of the 10 different amounts of solder at an integration time of 13 ms; C: Correspondence of camera integration time and detected keypoints on solder joints, background and its ratio; D: Correspondence of integration time and detection time t_{det} of FAST feature detection.

6.3.2 Feature Detectors and Parametrization

In this section the results of the proposed feature detectors at different parametrizations are presented. For every parameter set of a detector, the values of SolFeC, computation time and keypoint ratio were evaluated to determine the best detector with a corresponding parametrization for solder joint keypoint detection.

MSER For the MSER detector the Δ gray level was changed. A region becomes maximally stable at different thresholds, which means it remains a connected region during the change of thresholds. It is visible in figure 6.18 that a better coverage can be obtained for smaller Δ gray levels. The best obtained value is 0.77 for a Δ gray level of 1, which is valid only for single solder joints. However, the mean coverage parameter does not rise above 0.3, which is considered to be not sufficient (see figure A.20). Besides the unsuitable SolFeC the computation time between 1000 ms and up to 1950 ms for every image (see figure A.21 in the appendix) is to high for an online inspection system.

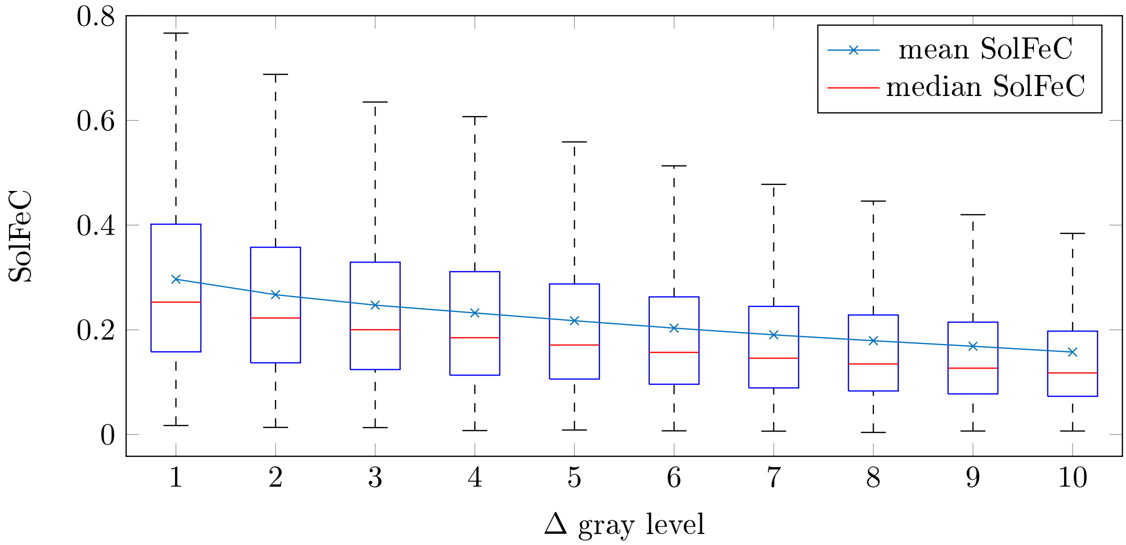


Figure 6.18 Coverage parameter of MSER in relation to Δ gray level (Δ gray level = 1 - 10)

SIFT Applying the SIFT detector requires the following 3 parameters to be set: number of octaves, contrast threshold and edge threshold. The resulting coverage of solder joints is given in figure 6.19. More octaves lead to a higher SolFeC as well as a lower contrast and a higher edge threshold. The detector reaches a maximum coverage of 0.25 which is not suitable. Using more than 5 octaves will not lead to a significant improvement, as one can expect from the graph trends in figure 6.19, where the increase of the SolFeC becomes smaller for higher octaves. For completeness the computation time can be seen in the appendix in figure A.23.

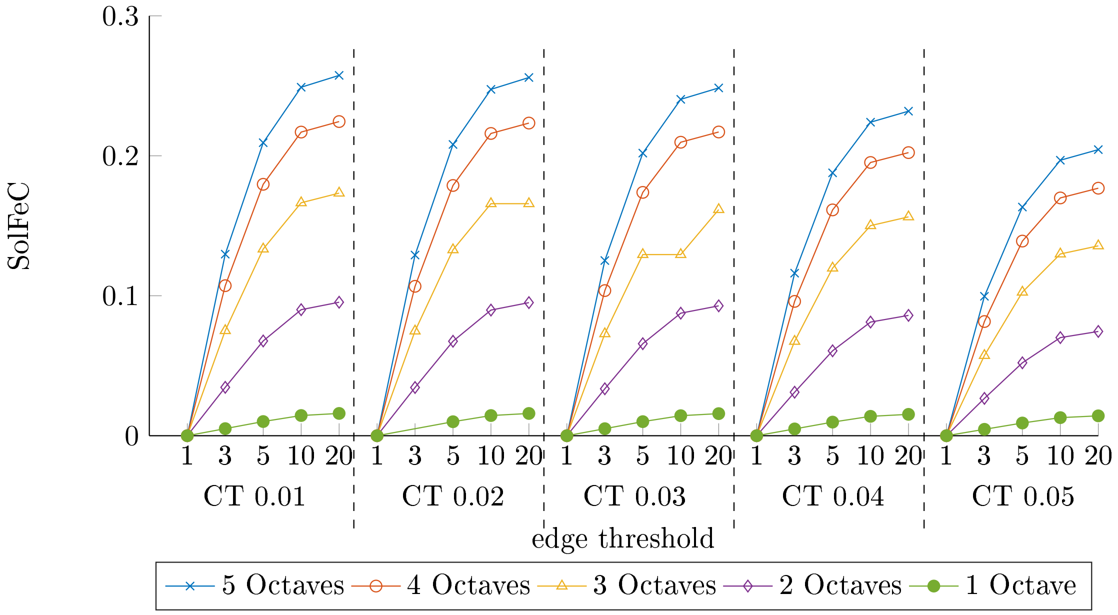


Figure 6.19 Solder joint feature coverage of SIFT keypoints for 5 different octaves, 5 contrast threshold (CT = 0.05, 0.04, 0.03, 0.02, 0.01) and 5 edge thresholds (20, 10, 5, 3, 1)

SURF The parametrization of the improved SIFT, the SURF detector, offers as much variations as SIFT, since 3 different parameters (number of octaves, layers per octave and Hessian threshold) are used. The detector was setup for 1, 2, 3 and 5 octaves (figure 6.20). The curves are split corresponding to the parametrization into 4 sections. Every octave-layer combination consists of 4 Hessian thresholds (1,100,300,500). The SolFeC decreases with a smaller number of layers per octave and a smaller number of octaves and with a larger Hessian threshold. 5 octaves, 20 layers per octave and a low Hessian threshold of 1 should be used to obtain the largest possible SolFeC parameter. However, even this configuration does not rise the coverage parameter above 0.11, which is insufficient for this application. Figure A.24 in the appendix shows the computation times for the detector configurations.

ORB For the ORB detector, two parameters were changed to identify a suitable keypoint detection setup. One to ten pyramid levels were used, where the contrast threshold was varied between 1 and 50. It is evident from figure 6.21 and 6.22, that a low contrast threshold leads to a high coverage value of over 0.9, if more than 3 pyramid levels for the detection were used. It can also be seen that the more levels are used, the higher the values of the coverage parameter are. However, the difference between the levels decreases with higher numbers of levels. Starting from 5 pyramid levels no significant increase with higher level values is obtained. A similar behavior is shown in figure 6.22, were the coverage distribution at 5 pyramid levels is plotted. A lower contrast value than 10 does not lead to a significant increase of the SolFeC. However, when comparing to

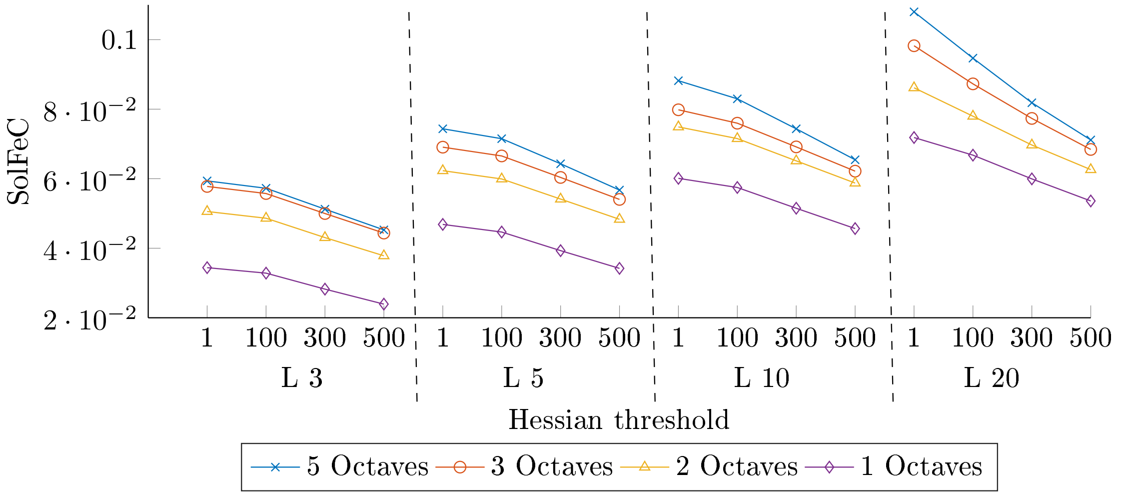


Figure 6.20 Solder joint feature coverage of SURF keypoints for 4 different octaves, 4 amounts of layers per octave (L = 20, 10, 5, 3) and 4 different hessian thresholds (500, 300, 100,1)

figures 6.23 and 6.24, it is obvious, that a further decrease of the contrast threshold leads to exponential increase of the computation time and background keypoints.

Additionally the detector was accelerated by CUDA. The computing time shows a low increase when lowering the contrast threshold from 50 (t_{det} = 11 ms) to 1 (t_{det} = 17 ms). A significant improvement in the time related performance is obtained and the detector is accelerated by a factor up to 20 considering 10 pyramid levels and a contrast threshold of 1, when comparing CPU-based (339 ms) and CUDA-based (17 ms) computation.

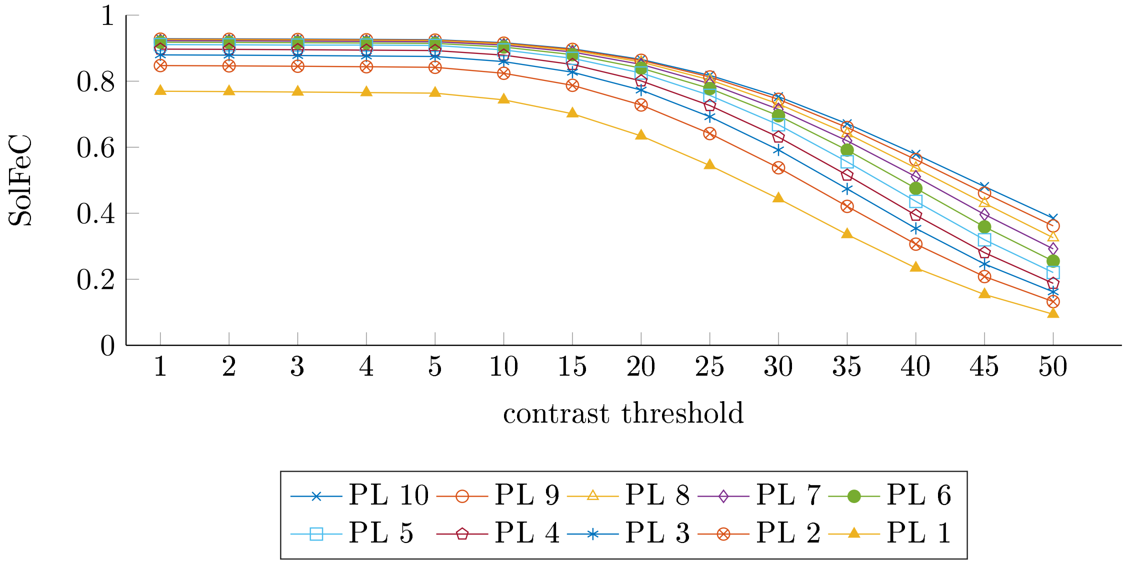


Figure 6.21 Solder joint feature coverage of ORB keypoints for 10 different amounts of image pyramid levels (PL = 1 to 10) and contrast thresholds 1 - 15 for each level.

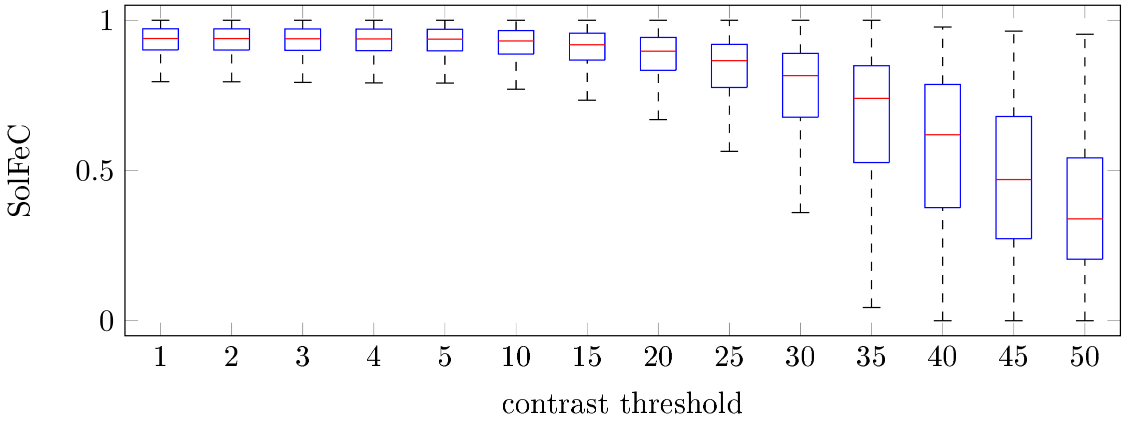


Figure 6.22 Distribution of keypoint coverage for the ORB detector using 5 pyramid levels across contrast thresholds 1 - 50.

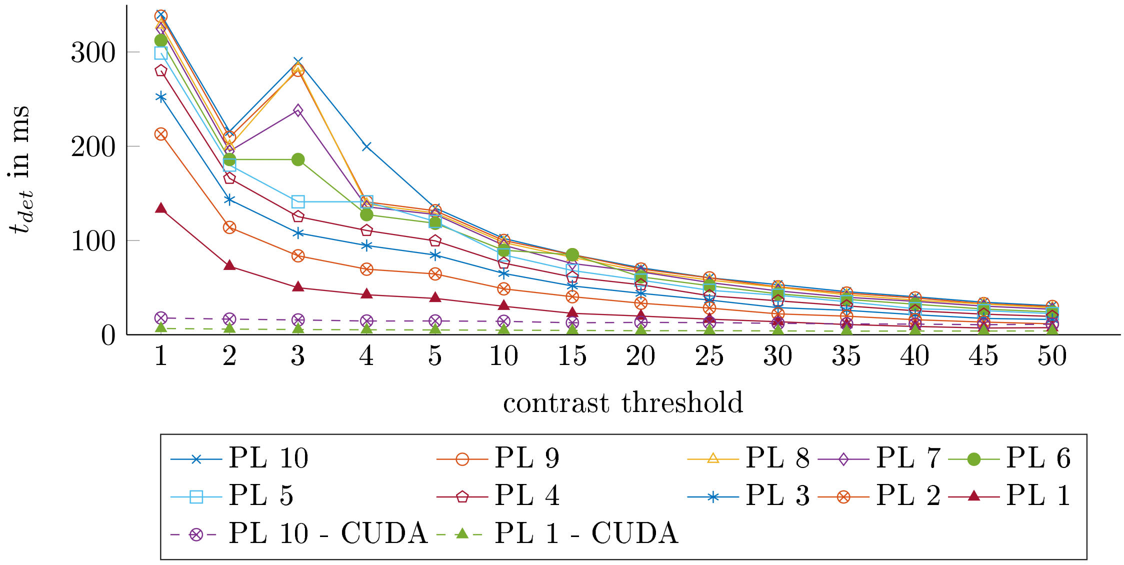


Figure 6.23 Computation time of ORB for 5 different pyramid levels with various contrast thresholds 1 - 50.

BRISK In figure 6.26 the influence of the detector parameter contrast threshold and number of octaves can be seen. Comparable to the previous detectors, a low contrast threshold leads to a high coverage. However, the increase of the coverage is less visible for higher number of octaves. Starting from 3 octaves the parameter curves are almost congruent, especially at lower contrast thresholds. A plot of the distributions of SolFeC parameters is given in figure 6.27. The parameter does not increase significantly for a threshold below 15. A further decrease of the threshold will lead to a strong increase of background keypoints (see figure 6.29), which also results in a significant

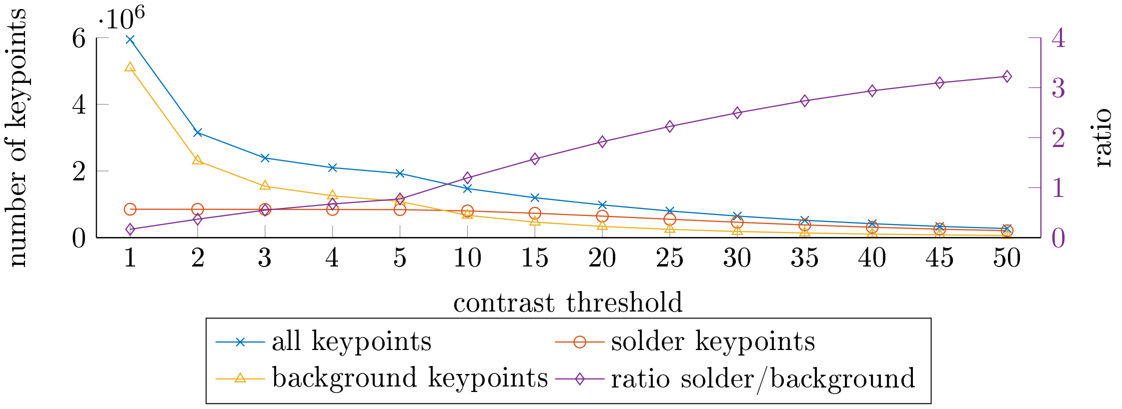


Figure 6.24 Keypoint detection of ORB; divided into keypoints of solder joints and background, total keypoint and solder/background keypoint relation for the examined amounts of 10 pyramid levels and different contrast thresholds.

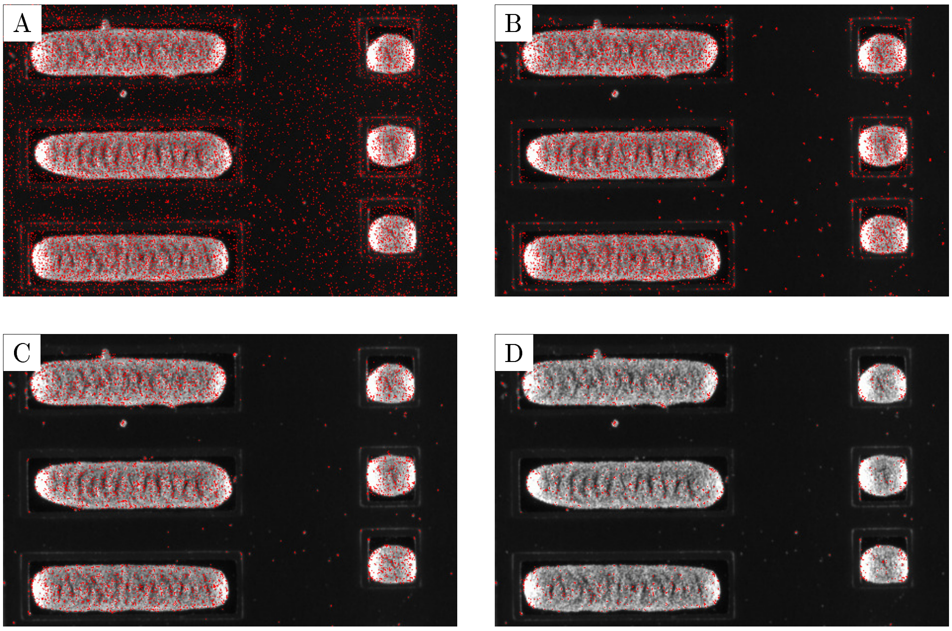


Figure 6.25 ORB - exemplary keypoint distribution; ORB feature detection was performed using 5 pyramid levels and A: contrast threshold = 1; B: contrast threshold = 10; C: contrast threshold = 25; D: contrast threshold = 50.

increase of the computation time at thresholds below 10 and even stronger below 5 as shown in figure 6.28.

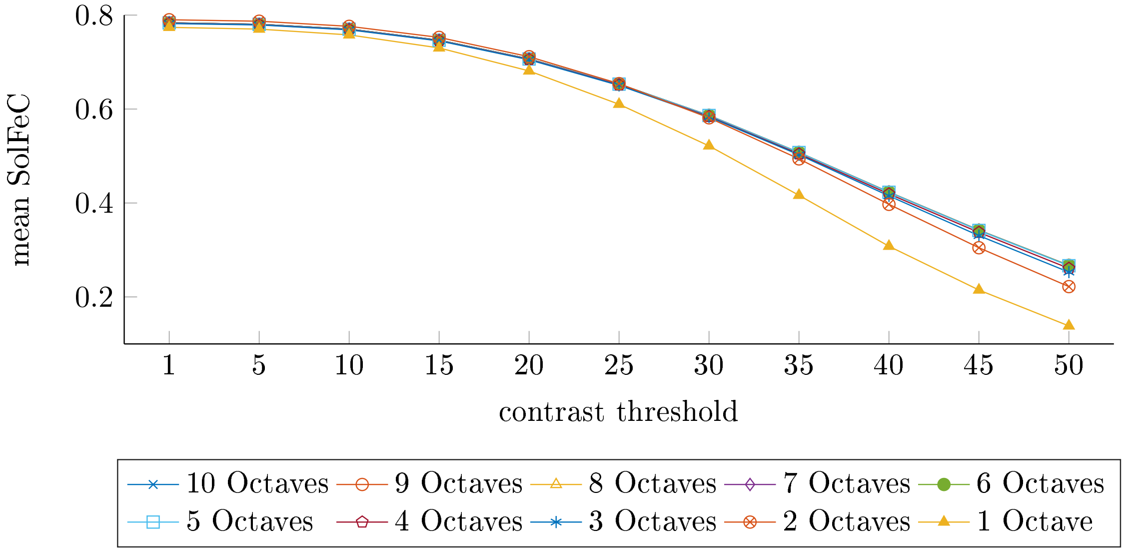


Figure 6.26 Mean coverage of solder joints with BRISK keypoints using 1 - 10 octaves and various contrast thresholds in each octave configuration (CT = 1, 5 ,10, 15, 20, 25, 30, 35, 40, 45, 50).

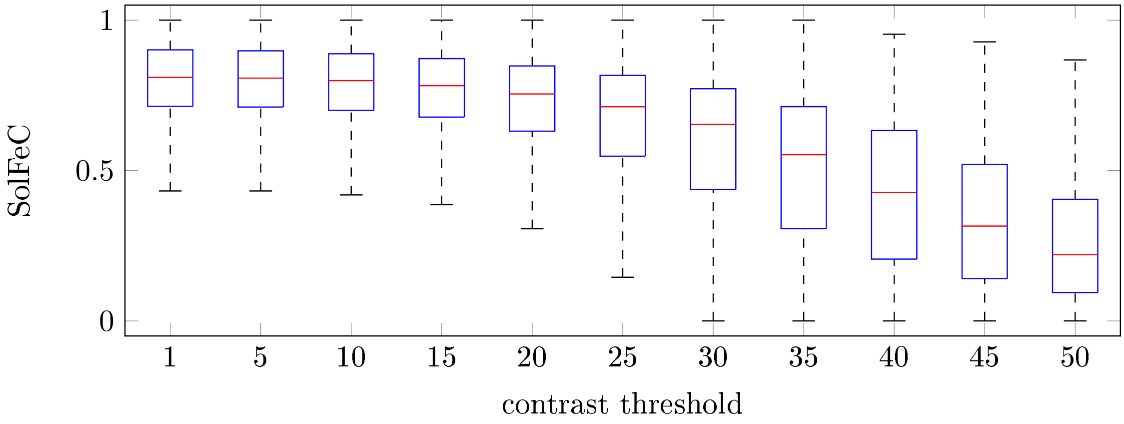


Figure 6.27 Distribution of SolFeC for the BRISK detector using 10 octaves across the examined contrast thresholds (CT = 1, 5 ,10, 15, 20, 25, 30, 35, 40, 45, 50).

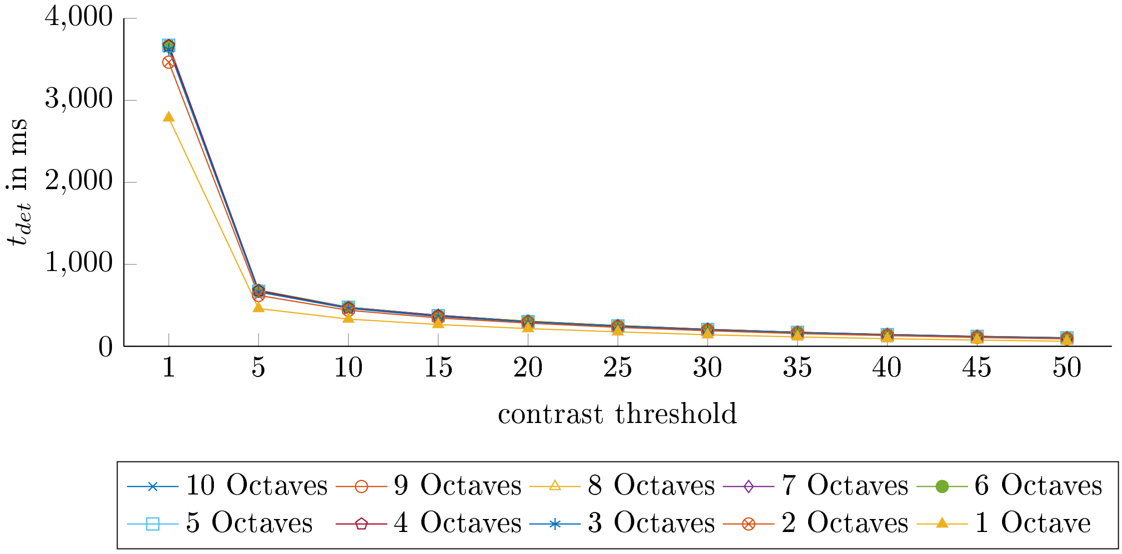


Figure 6.28 Computation time of BRISK for up to 10 different octaves and the contrast thresholds CT = 1, 5 ,10, 15, 20, 25, 30, 35, 40, 45, 50.

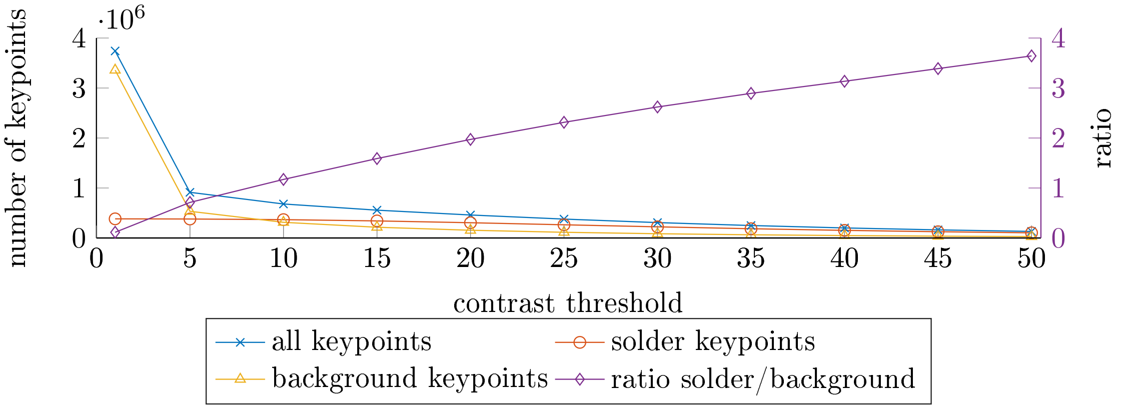


Figure 6.29 Keypoint detection of BRISK; divided into keypoints of solder joints and background, total keypoint and solder/background keypoint ratio for 10 octaves and contrast thresholds CT = 1, 5 ,10, 15, 20, 25, 30, 35, 40, 45, 50.

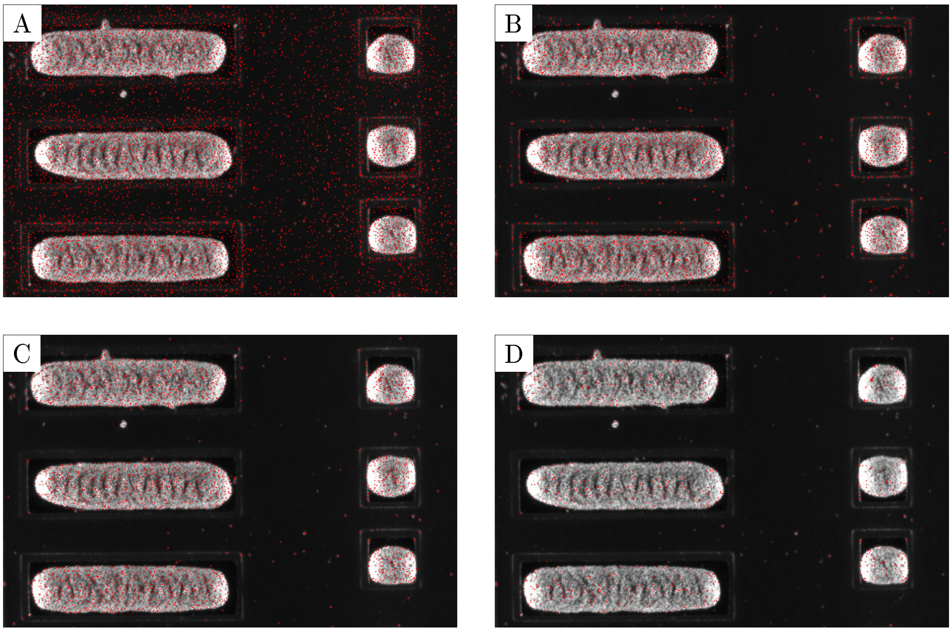


Figure 6.30 BRISK - exemplary keypoint distribution; BRISK feature detection was performed using 10 octaves and A: contrast threshold = 1; B: contrast threshold = 10; C: contrast threshold = 25; D: contrast threshold = 50.

FAST The FAST detector offers only one adjustable parameter; the contrast threshold. Similar to other detectors, a low contrast threshold leads to a higher amount of detected keypoints. This is true for solder joint and non solder joint regions as visible in figure 6.33 with an exponential like increase of computation time (see figure 6.32). However, this increase is not able to raise the solder joint feature coverage significantly for values below 10 as it is shown in figure 6.31. This detector was also used with an hardware acceleration using NVIDIA's CUDA. From figure 6.32 the potential of this acceleration becomes evident. The detector becomes up to 10 times faster on the GPU, with maximum detection times of 2 ms and a minimum detection time of 0.05 ms. Along the contrast thresholds no significant behavior of the time consumption is apparent. Due to these results, a contrast threshold of 5 delivers the best results in terms of the SolFeC parameter and keypoint ratio. Table 6.5 shows the resulting best parametrization of each feature detector. The parameters were chosen to obtain a high SolFeC parameter and a short computing time t_{det}.

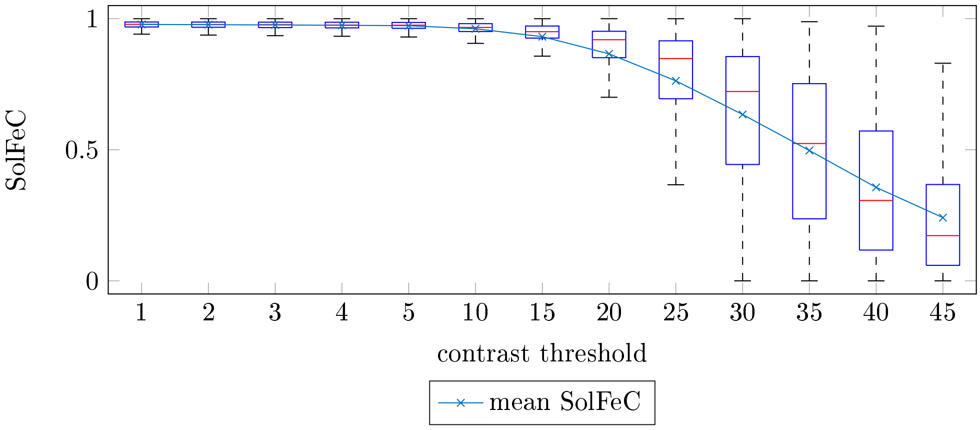


Figure 6.31 Solder joint feature coverage of FAST keypoints for the contrast thresholds CT = 45, 40 35, 30, 25, 20, 15, 10, 5, 1

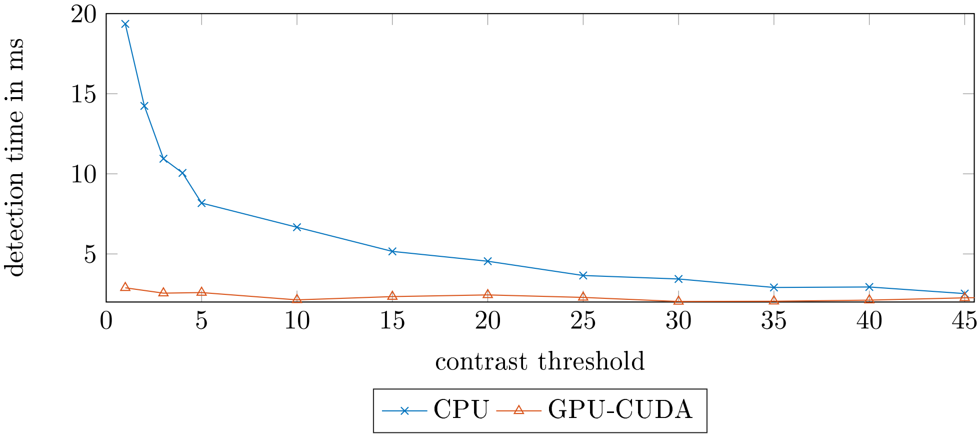


Figure 6.32 Computation time of FAST for the contrast thresholds CT = 45, 40 35, 30, 25, 20, 15, 10, 5, 1

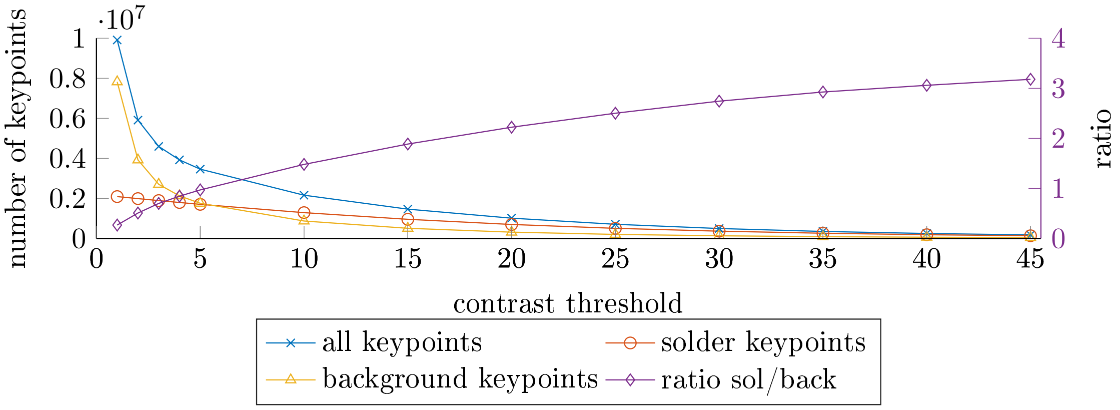


Figure 6.33 Keypoint detection of FAST; divided into keypoints of solder joints and background, total keypoint and solder/background keypoint relation for the examined contrast thresholds (CT = 45, 40 35, 30, 25, 20, 15, 10, 5, 1).

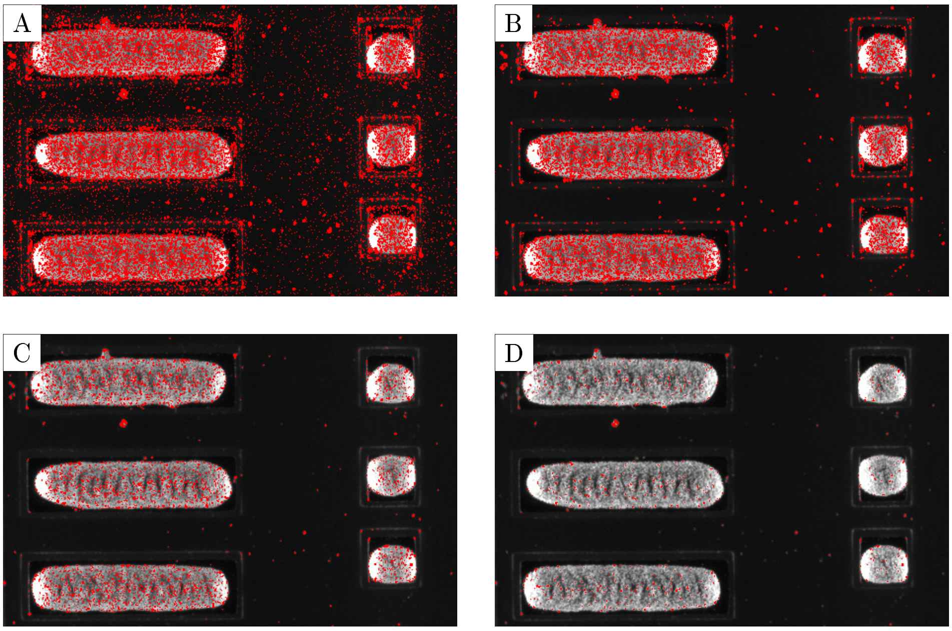


Figure 6.34 FAST - exemplary keypoint distribution; FAST feature detection was performed using A: contrast threshold = 1; B: contrast threshold = 10; C: contrast threshold = 25; D: contrast threshold = 45.

The examination of the detection results yield the FAST and ORB detector as the best detectors for the task of this thesis. Especially its GPU-based implementations lead to appropriate maximum computation times of 2 ms. A high SolFeC parameter of 0.93 and 0.98 was reached, where the number of solder joint keypoints is 3 times higher compared to the number of background keypoints. Other detectors were not able to produce a SolFeC parameter higher than 0.3 and had computation times of up to 2 seconds. The exception is BRISK with a SolFeC of 0.77 but with 469 ms of

Table 6.5 Best parametrization of feature detectors, CT - contrast threshold, ET - edge threshold, HT - Hessian threshold, PL - pyramid levels, O - octaves, L - layers per octave

detector	SolFeC	t_{det} in ms	parametrization
MSER	0.3	1950	Δ graylevel = 1
BRISK	0.77	469	CT = 10, O = 4
SIFT	0.25	850	ET = 20, CT = 0.03, O = 5
SURF	0.11	790	L = 20, HT=1, O = 5
FAST	0.98	8.5	CT = 10
ORB	0.93	102	PL = 10, CT = 10
FAST - GPU	0.98	2.5	CT=10
ORB - GPU	0.93	14.2	PL = 10, CT = 10

computation time, which is inappropriate for online feature detection. Thus, FAST and ORB were used for further processing within the feature matching.

6.4 Feature Matching

6.4.1 Computation Time

The matching methods were compared according to the computation time, recall and precision or 1-precision. Figure 6.35 shows the computation time of all combinations of ORB/FAST detectors from table 4.2 and the descriptors from section 2.4.6. The detection was performed on a data set containing 100% solder pad coverage (as shown in figure 4.5). The plot contains only the time that is used for the matching to be finished, not the feature detection. The computation time is highly dependent on the combination of detector and descriptor. The combinations can be compared directly since the two detectors yield a similar amount of feature points of 2,000,000 for the configuration in table 6.5. Especially the usage of SURF and SIFT descriptors in combination with a brute force L1 or L2 norm yield a large matching time of up to 90 seconds, which is inappropriate for an online inspection system. These descriptors are followed by BRISK and BRIEF. In any case the FLANN based matching delivers the best timing results when the matching is performed using no GPU (CUDA) acceleration. In general, using FLANN or GPU accelerated matching methods result in a small computation time. Note that the hamming distance based matching is missing for the SURF and SIFT descriptors, since these are no binary descriptors and the hamming distance, as defined in table 2.3, cannot be calculated. Again SIFT descriptors yield the largest processing time and ORB the shortest followed by the usage of BRIEF. The maximum processing time is decreased to 1.35 seconds when using CUDA-SIFT as the descriptor. A minimum matching time of 300 ms at a maximum feature point number is obtained by using the ORB algorithm for feature detection and description within a hamming distance based matching.

6.4.2 Recall and Precision

Recall and precision measures give a statement on the quality of a matching method. The number of resulting matches is adjusted by setting a threshold to the vector distance (L1/L2, FLANN and Hamming) of feature descriptors. This distance is also called the matching score. Allowing all distances, every feature point gets a match. However, this results also in false matches. Lowering the threshold yields more strong matches and less false matches but also to feature points without a match. By adjusting the threshold of the allowed vector distance, the precision-recall curves in figure 6.36 and figure 6.37 are obtained. A high recall value means that many point correspondences obtain a match, whereas a high precision value states the correctness of found matches. Every combination of detector/descriptor/matching method was evaluated. In figure 6.36 the recall-precision curves of the descriptors BRIEF (A,D), BRISK (B,E) and ORB (C,F) in combination with the FAST detector (left column) and the ORB detector (right column) are shown. FLANN and L2 are almost congruent and yield the worst performance in terms of recall and precision, followed by L1. The best result is given by the hamming based matching for all combinations. Figure 6.37 shows the results for SURF (A,C) and SIFT (B,D) in combination with the FAST

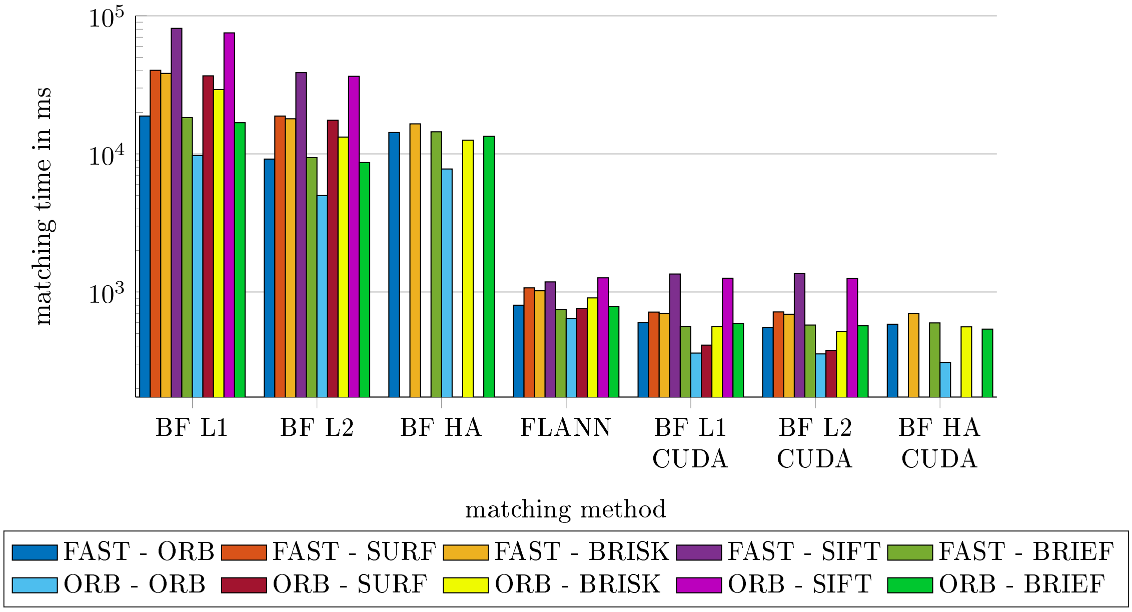


Figure 6.35 Computation time for feature matching using 100% solder pad coverage (see figure 4.5); BF - Brute Force; L1/L2 - vector distance using L1/L2 norm; FLANN - Fast Library for Approximate Nearest Neighbor Search; HA - hamming distance based matching. The combination indicate the use of a detector algorithm and a descriptor algorithm. Example: FAST-ORB: FAST detector and ORB descriptor.

detector (left column) and the ORB detector (right column). The trend of the curves is similar. L1 delivers the best result followed by L2 and FLANN. An increase of the recall mainly leads to an decrease of precision with the exception of the ORB/BRIEF and FAST/BRIEF (plots A and D in figure 6.36) where a local minimum can be found. A similar behavior can be found for the L1 curve in the FAST/SIFT combination (plot B in figure 6.37). All combinations show a best recall-precision curve. These are summarized in figure 6.38 to allow for cross comparison. It is evident, that a L1-norm matching for a FAST/SIFT and ORB/SIFT combination deliver the best results. These combinations are followed by the hamming distance based matching of the ORB/BRISK combination. The worst recall-precision values are given by the FAST/SURF/L1 combination. Figure 6.39 shows the number of correct matches of the algorithm combinations in figure 6.38. FAST/SIFT clearly delivers the highest number of correct matches for all matching methods (except for the hamming distance) with approximately 1300 correct matches. In contrast, FLANN based matching yields the lowest amount of correct matches. If hamming is used, the most matches are obtained by the FAST/BRIEF combination with 820 correct matches, followed by the FAST/ORB combination with 680 correct matches and 560 correct matches are established by the ORB/BRISK combination. Since it is intended to use non-patented algorithms that are as fast as possible, the combination FAST/BRIEF and a hamming based matching is used in the next section.

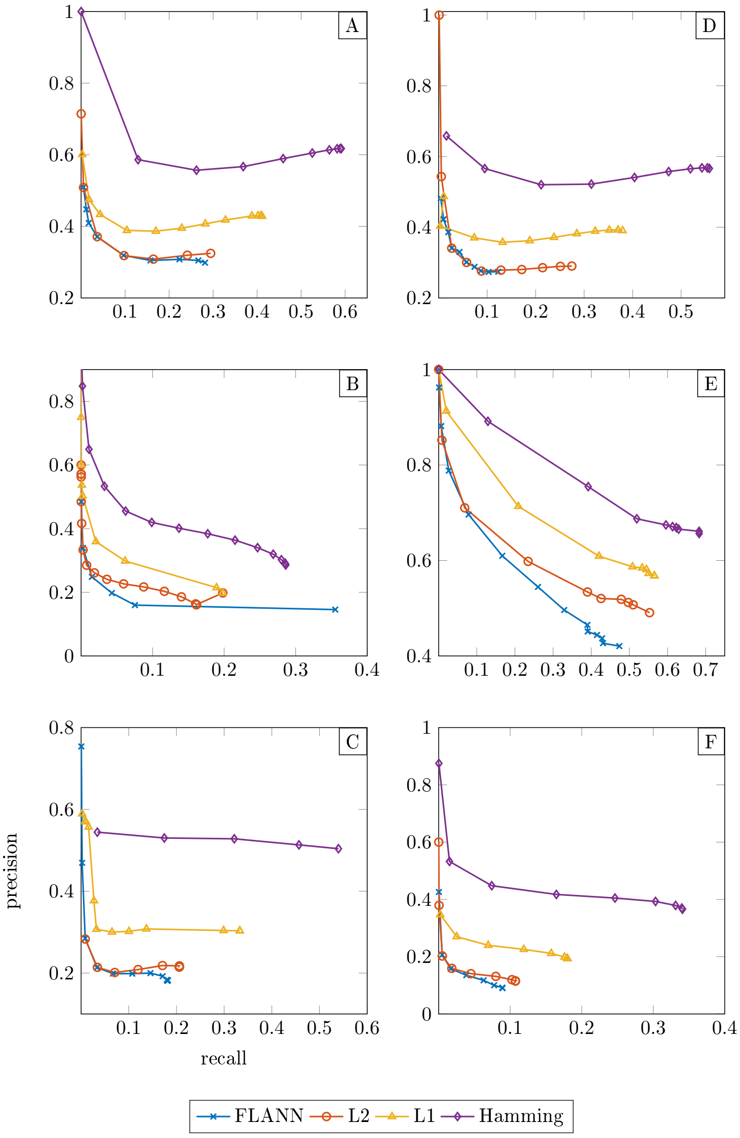


Figure 6.36 Precision recall curve for different combinations of detector/descriptor - A: FAST/BRIEF; B: ORB/BRIEF; C: FAST/BRISK; D: ORB/BRISK; E: FAST/ORB; F: ORB/ORB.

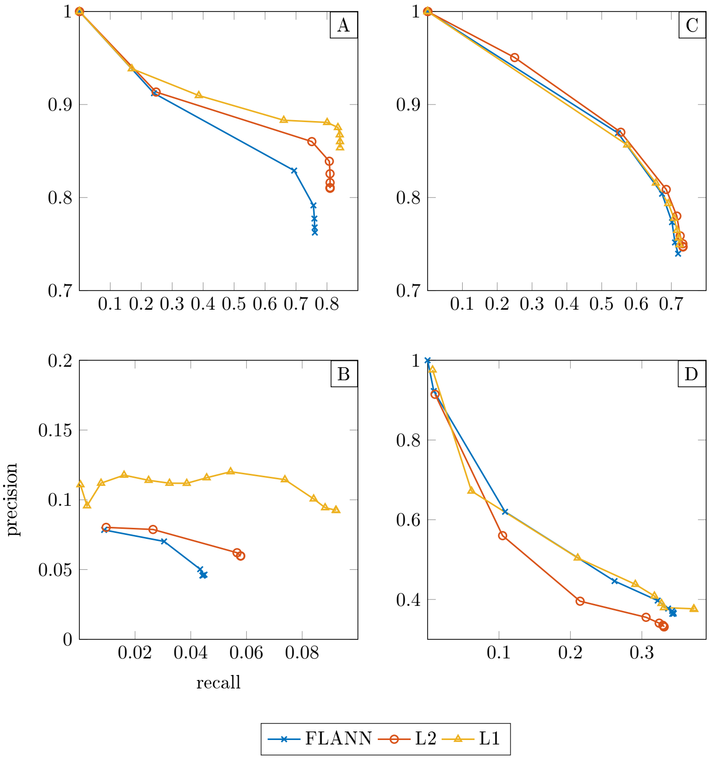


Figure 6.37 Precision recall curve for different combinations of detector/descriptor(continued) - A: FAST/SIFT; B: ORB/SIFT; C: FAST/SURF; D: ORB/SURF.

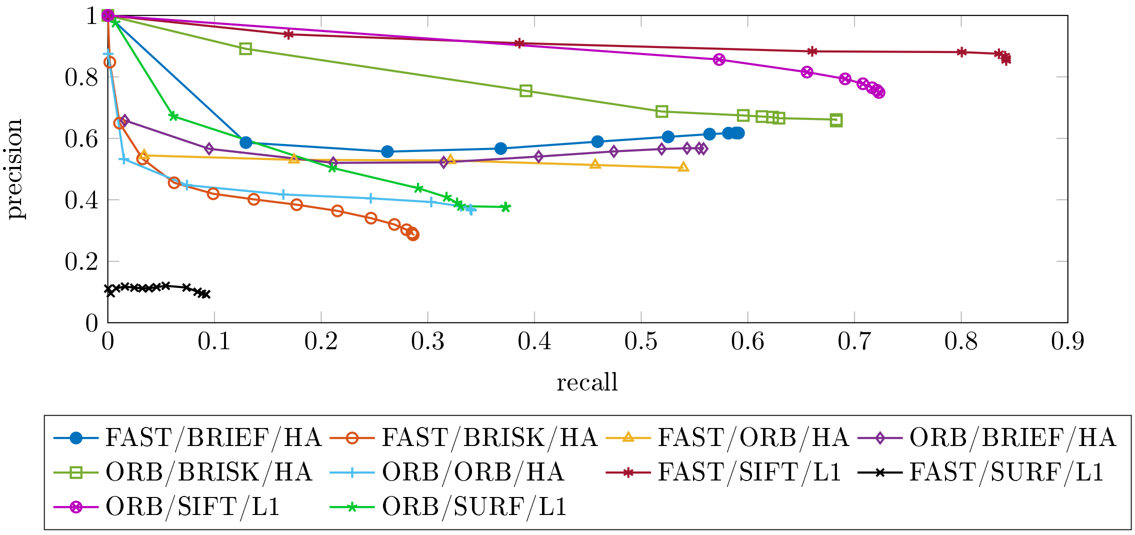


Figure 6.38 Best precision recall curves. BRIEF, BRISK, ORB combinations are obtained by the hamming distance, whereas SIFT and SURF are obtained by the L1 norm. The results are based on the best detector configurations from table 6.5 and a 100% solder pad coverage.

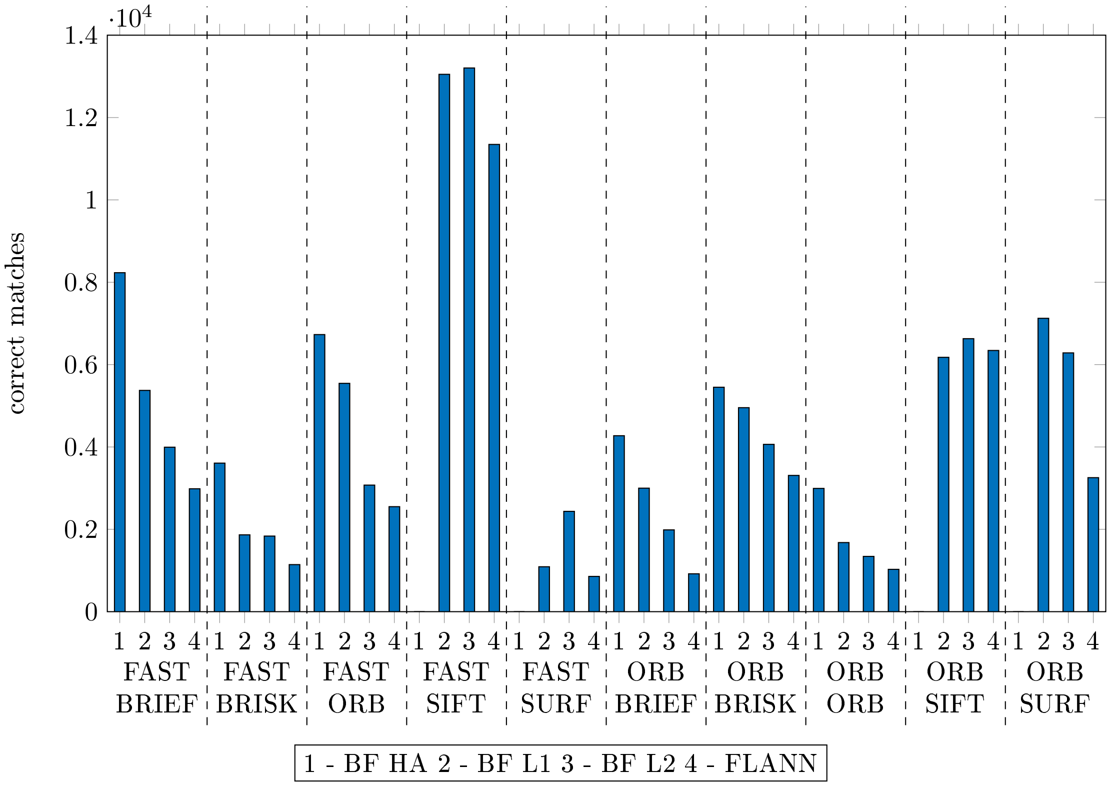
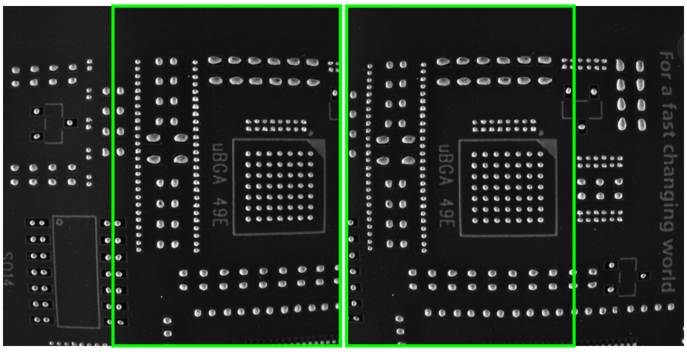


Figure 6.39 Number of correct matches for different detector/descriptor/distance measure combinations.

6.4.3 Matching Correction

The two exemplary images in figure 6.40 result from a horizontal camera movement and are used to visualize the result of the matching and its correction. The green rectangles indicate overlapping image areas where correct matches can be established. Due to the horizontal movement of the camera correct matches set up horizontal lines. The detected feature points were matched using a hamming distance based brute force matching. Without considering the matching score (descriptor vector distance) and any correction method many false matches are obtained, due to the homogeneous appearance of solder joints. This can be seen in figure 6.40 (bottom) by the confuse ordered lines across the images connecting also feature points outside the overlapping areas. The result of the first investigated correction method is shown in figure 6.41, where matches are eliminated by the vector distance. It can be seen that a large amount of false matches is removed by a small vector distance, resulting in a large precision value. This in turn leads to a low recall and less total matches (see image A in figure 6.41). When increasing the recall by rising the distance threshold more matches are obtained (see B and C in figure 6.41). This method has only limited capabilities to remove false matches whereas preserving correct matches, since the descriptors are very similar to each other. The result of the matching correction using the fundamental matrix based approach can be seen in figure 6.42. The horizontal matching lines indicate the correct consideration of the acquisition geometry. Only matches, that correspond to the epipolar geometry remain. However, also false matches considered to be true since they match the epipolar lines, caused by the normal case imaging setup. This can be seen for example by the matching lines that leave the overlapping areas indicated green. Using the camera models to eliminate false matches results in the most suitable correction of these three methods as it can be seen in figure 6.43. The method takes the exact projection of 3D points into account. Thus, matches that do not fit both camera models (one model for each image) within the mentioned tolerance are removed. It becomes evident, that only horizontal matches within the overlapping regions do remain. The matches are aligned to the camera shift direction and establish correct point to point correspondences. After the determination of correct matches, the pipeline (figure 1.6) proceeds to the triangulation . Depending on the result of matching correction the point cloud reconstructions in figure 6.44 are obtained.



camera view I camera view II

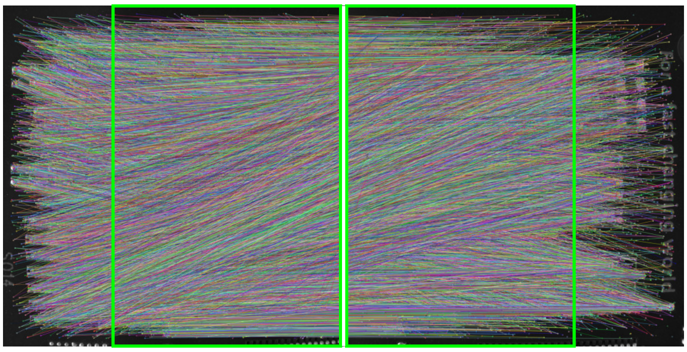
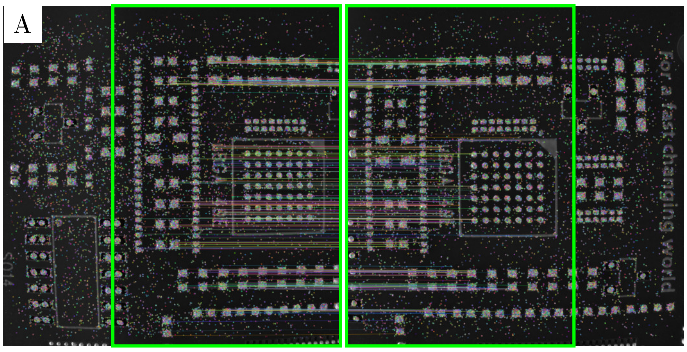
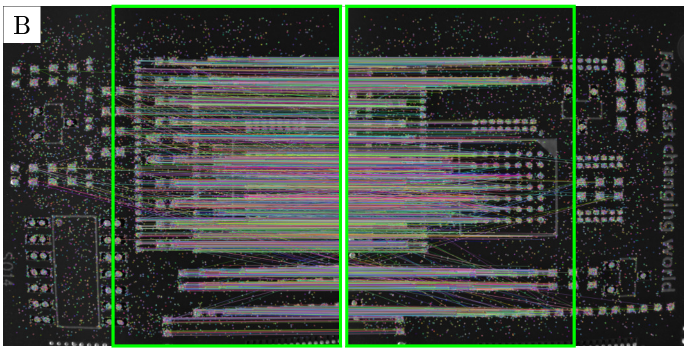


image I image II

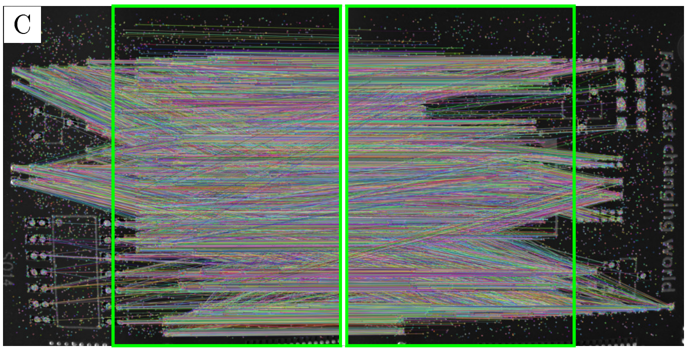
Figure 6.40 Exemplary input images (top) for two camera views (views are obtained by moving the camera) and feature matching without correction (bottom). The green rectangles indicate the overlapping regions in the two images were relevant feature points are located. Colored lines visualize matches between the left and right image.



recall = 0.12916 precision = 0.8916



recall = 0.39202 precision = 0.75473



recall = 0.51943 precision = 0.68739

Figure 6.41 Vector distance based correction: depending on the vector distance matches are removed. A high precision and low recall value yield the upper correction result. The distance threshold is increased from top to bottom, which results in a higher recall and decreased precision. Thus, the number of matches increases from top to bottom. However, these matches contain more false than correct positives.

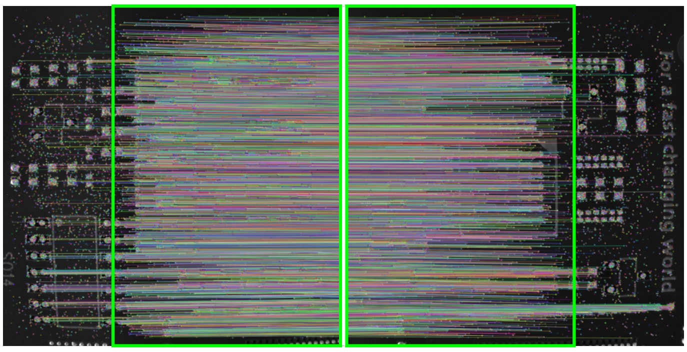


image I image II

Figure 6.42 Fundamental matrix based correction: Only matches (colored lines) that satisfy the condition of epipolar geometry do remain as correct matches. Since the condition becomes true for multiple points due to parallel epipolar lines, also some false matches are set to true.

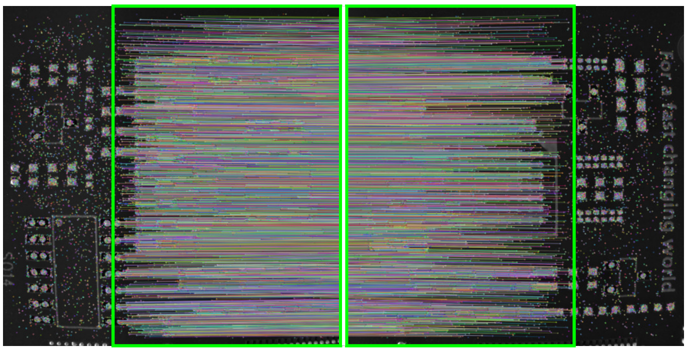


image I image II

Figure 6.43 Camera model based correction; The consideration of 3D projections within the corresponding camera models lead to the elimination of false matches. Only matches that fit both camera models remain.

6.5 3D Reconstruction, Point Cloud Processing and Volume Measurement

After the determination of correct matches, the pipeline proceeds to the 3D processing steps. The first step is the triangulation. Due to the result of matching correction a precise point cloud reconstruction is obtained, when using the camera model based approach. As it can be seen in figure 6.44, the reconstructed point cloud can contain a large amount of 3D points, that are non-existent in reality, if the matching correction is non sufficient.

Triangulation Figure 6.44 shows how false matches influence the reconstruction. The first row shows a top (A) and perspective (B) view of the reconstruction if no matching correction is applied. The cloud contains a large amount of false 3D points. The vector distance based correction, using 10% strongest matches can be seen in the second row with picture C (top view) and D (perspective view). The structure of the PCB becomes clearly visible and the position of solder joints can be extracted. However, the point cloud is very sparse and still contains false 3D points. The reconstruction using the correction by the fundamental matrix approach is represented by the third row. The PCB and its solder joints become rarely visible in the top view (E). The perspective view (F) shows false 3D points above the PCB. The last row shows the top view (G) and perspective view (H) of a reconstruction after the camera model based correction. The top and perspective views show a representation of the PCB surface and solder joint positions, where no false 3D points are visible above the PCB.

Clustering and Ground Projection The problem of clustering becomes clear in figure 6.45. Performing the clustering with and without a previous back-projection of detected solder joints onto the PCB plane (image A - without ground projection; image F - with ground projection) results in not usable and precise point cloud clusters (see image B and G). As can be seen, the solder joints are divided into multiple clusters if no ground projection and a short cluster tolerance is used (image D). The cluster tolerance needs to be increased to obtain solid clusters. However, that leads to unwanted connections between solder joints with a small distance to each other (image C and E). The green region in image E represents a single Cluster, which consists of multiple joints. Thus, it clarifies the absolute necessity of back projection of segmented solder joints. It can be seen that solder joints cluster become solid when using the ground projection, whereas different solder joints remain separated (image H).

Concave Hull Vertices Figure 6.46 shows the result of the extraction of concave hull vertices. For visualization reasons a zoom to a selected solder joint was performed. The left part (A) of the figure shows the cluster of this solder joint containing its projected ground points (dense plane) and the distributed points from the triangulation above the ground plane. The calculation

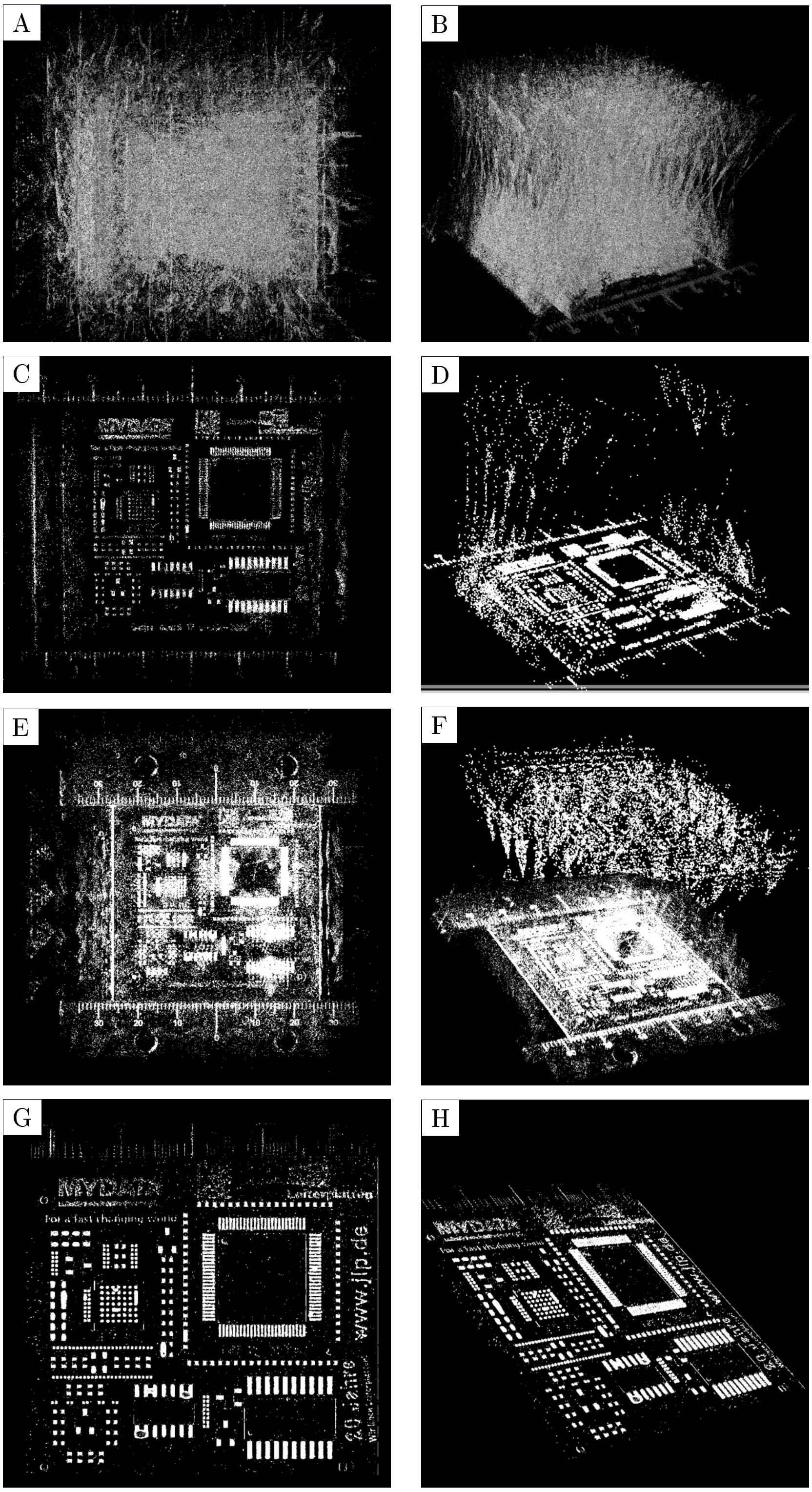


Figure 6.44 Point cloud results of triangulation of feature points after matching correction - left column: top view of reconstructed point cloud, right column: perspective view of the same cloud; Clouds are obtained from triangulation after different matching corrections: A,B: no correction, volume was cut to isotropic 50 mm cube for visualization; C,D vector distance correction, top 10% of matching scores are used. E,F: fundamental matrix correction; G,H: camera model based correction.

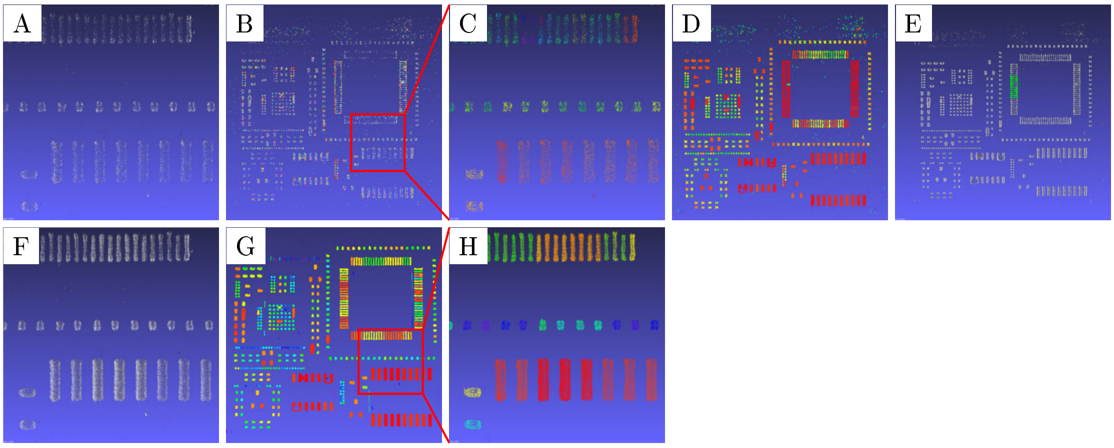


Figure 6.45 Euclidean clustering of reconstructed point cloud: A,F - enlarged PCB region without (A) and with (F) ground projection; B - resulting clusters from input of A; G - resulting clusters from input of F; C,H - enlarged cluster region, in C a large distance (100 micrometer is used to define solid joints, which is still not sufficient as it can be seen by the divided joints at the top, whereas a small distance (20 micrometers) in H is sufficient using ground projected points. Further decrease of cluster tolerance (150 micrometers) yields solid joints in D but also yields connected joints as displayed in E (green region).

of hull vertices removes points from the inside of the solder joint. The right part (B) shows the remaining hull vertices. All 3D points within a single solder joint cloud are removed.

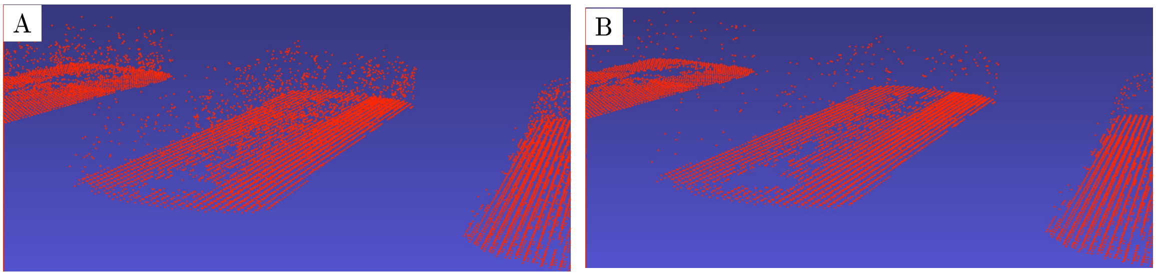


Figure 6.46 Concave Hull Vertex Points - A: input cluster, B: extracted concave vertex points.

Surface Reconstruction and Volume Measurement In figure 6.47, the final surface reconstruction can be seen. The hull vertices are triangulated and smoothed. From these closed surfaces the volumes were extracted as explained in section 4.3.3. A total of 480 solder joint volumes were calculated and evaluated by a ground truth, which was obtained by a structured light measurement (same characteristics like mentioned in the 3D target measurement) of the PCB. Data set 3 from section 4.1.3 (double sided illumination, clean PCB surface) was used.

In figure 6.48 a comparison of the estimated volumes from the developed software and the structured light in combination with Poisson surface reconstruction (software: AutoDesk Civil 3D) is shown. Matching volumes are at the diagonal line between minimum an maximum volume. Deviations in the volume estimation are expressed by points that differ from the line. It can be

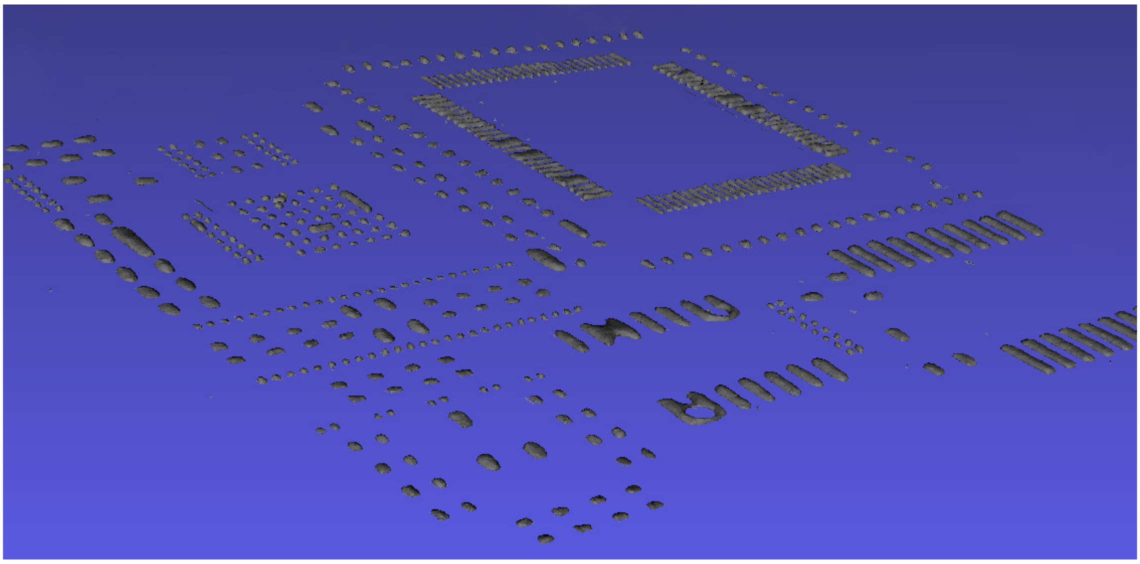


Figure 6.47 Delaunay based surface reconstruction

seen, that the absolute deviations are comparable for solder joints of different volumes, with the exception of volumes smaller than 10 nl, where the deviation is smaller. Tables 6.6 and 6.7 give an overview over the distribution of the absolute and relative deviation, respectively. For better comprehensibility, the volumes are structured into 5 classes according to their volume. As it can be seen, the mean absolute deviation increases with the solder joint volume (see table 6.6), whereas the mean relative deviation decreases for larger solder joint volumes as it is shown in table 6.7.

To show this more in detail figure 6.49 visualizes the positive and negative deviation and figure 6.50 the relative deviation in box plots, again for sub classes of solder paste volumes. It can be seen that the solder volume is overestimated compared to the ground truth for most of the solder joints.

Table 6.6 Absolute solder joint deviation

volume	up to 10 nl	11-25 nl	30-75 nl	75-150 nl	150-200 nl
median	1.01	1.98 nl	5.03 nl	7.51 nl	4.96 nl
mean	1.15 nl	3.09 nl	5.33 nl	5.78 nl	7.95 nl
σ	1.13 nl	3.01 nl	4.40 nl	5.80 nl	6.90 nl

Table 6.7 Relative solder joint deviation

volume	up to 10 nl	11-25 nl	30-75 nl	75-150 nl	150-200 nl
min in %	0.1	0.16	0.04	0.12	0.09
max in %	61.77	65.79	38.87	12.17	10.52
mean in %	16.93	16.41	11.61	5.31	4.55

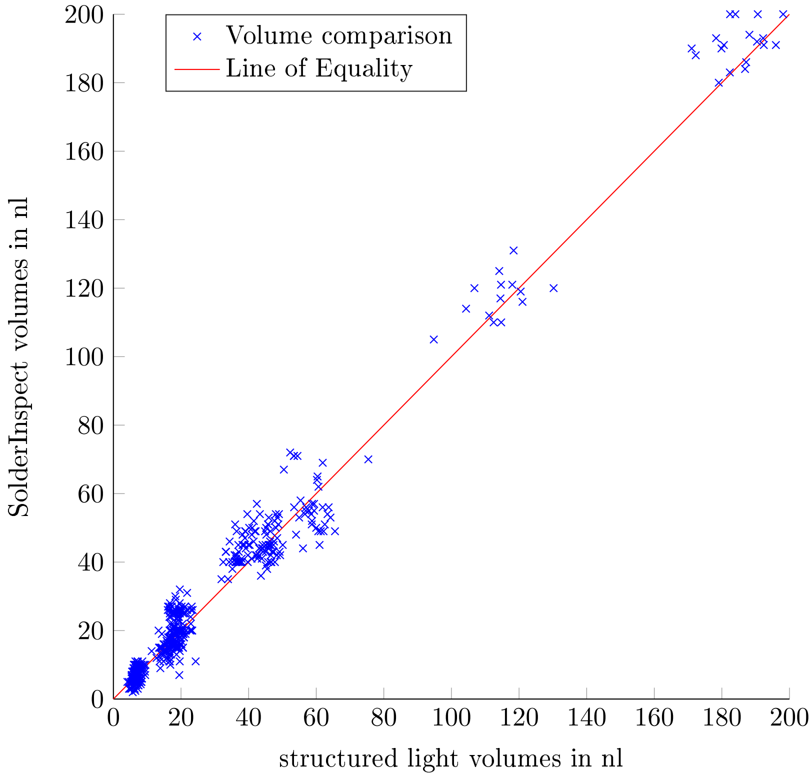


Figure 6.48 Volume Estimation - Comparison of volumes estimated by structured light and SolderInspect; every blue dots represent the volume of a solder joint measured by SolderInspect (abscissa) and the volume of the same solder joint measured by the structured light inspection system (ordinate). If the values are equal the dot lies on the line of equality (red).

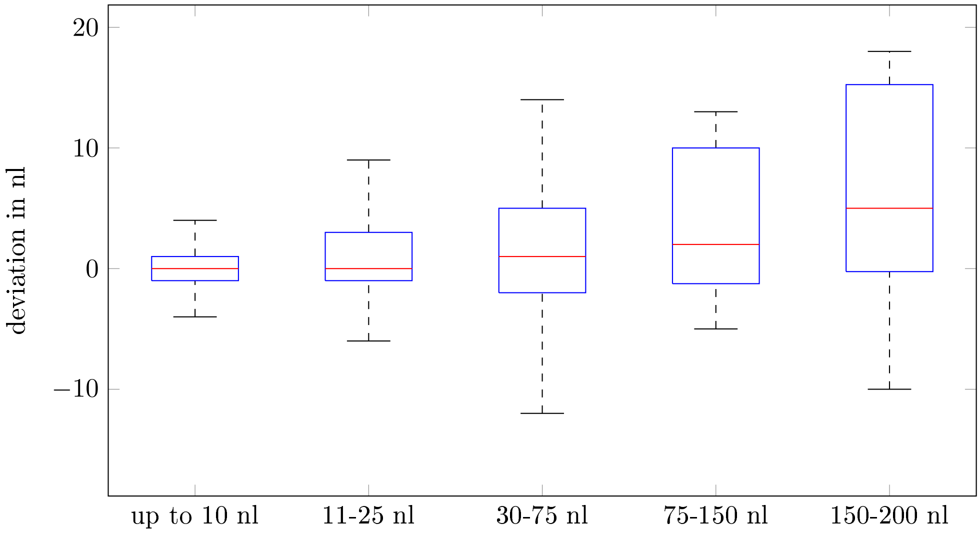


Figure 6.49 Volume estimation - Absolute deviation of volumes estimated by structured light and SolderInspect. The Deviation is divided by the amount of solder into 5 classes: up to 10 nl, 11-25 nl, 30-75 nl, 75-150 nl, 150-200 nl.

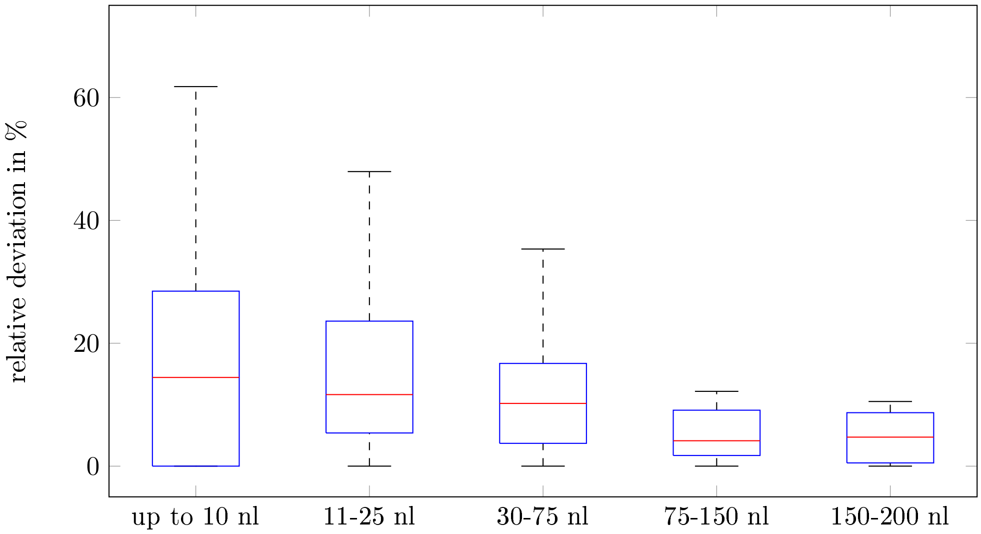


Figure 6.50 Volume estimation - Relative deviation of volumes estimated by structured light and SolderInspect. The Deviation is divided by the amount of solder into 5 classes: up to 10 nl, 11-25 nl, 30-75 nl, 75-150 nl, 150-200 nl.

In figures 6.51 the relative deviation of the volume measurement is shown for all solder joints. In figure 6.52 the histogram is accumulated into 12% bins, which corresponds to the required accuracy. Thus, the first bin corresponds to acceptable estimations. Calculating the mean deviation over all solder joints a value of 14% is obtained. Finally the repeatability of the volume estimation was examined. Therefore, the whole processing chain (including image acquisition) was performed 50 times on the same test PCB. It can be seen in figure 6.53 and table 6.8 that the deviation of the volume estimation in terms of repeatability decreases with an increase of the volume.

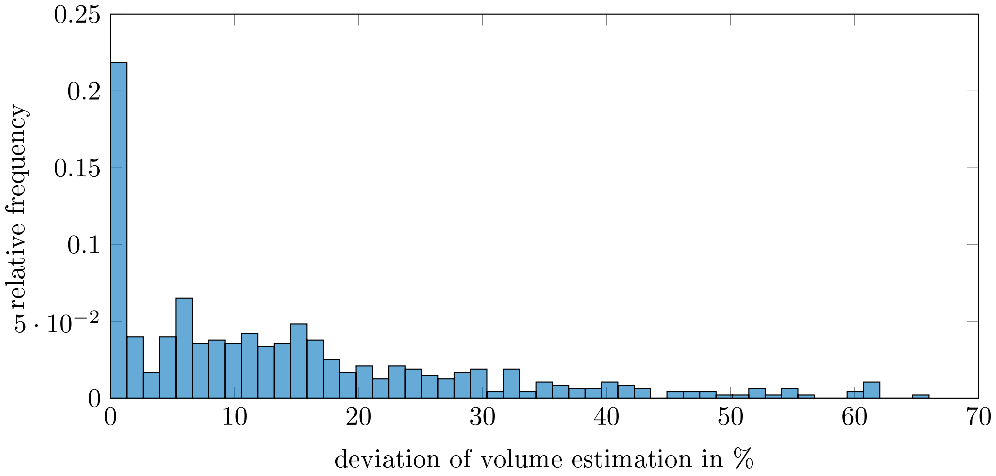


Figure 6.51 Volume Estimation - Histogram of relative deviation.

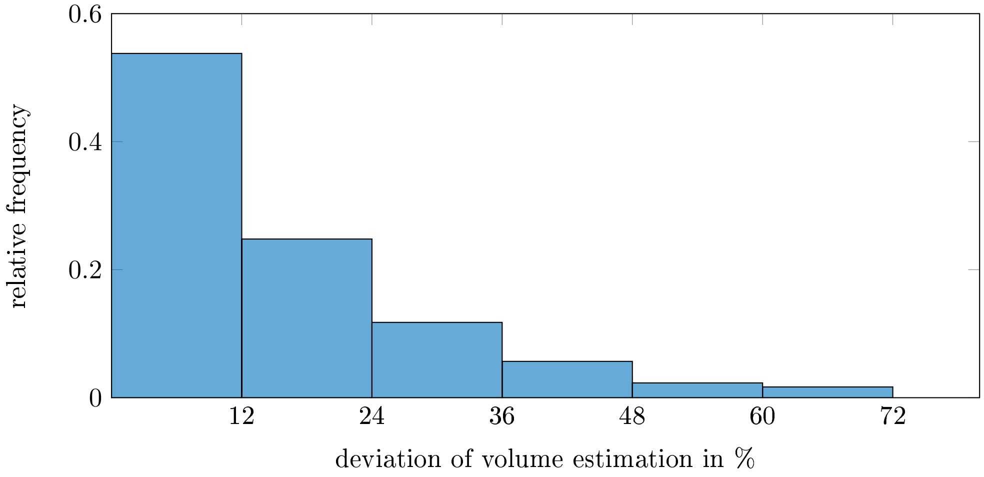


Figure 6.52 Volume Estimation - Accumulated histogram of relative deviation using 12% bins, corresponding to the required accuracy.

Table 6.8 Repeatability: relative solder joint deviation

volume	up to 10 nl	11-25 nl	30-75 nl	75-150 nl	150-200 nl
min in %	3.95	3.72	1.79	0.73	0.91
max in %	16.57	16.06	7.76	5.15	3.24
mean in %	9.96	8.43	5.82	2.91	2.49

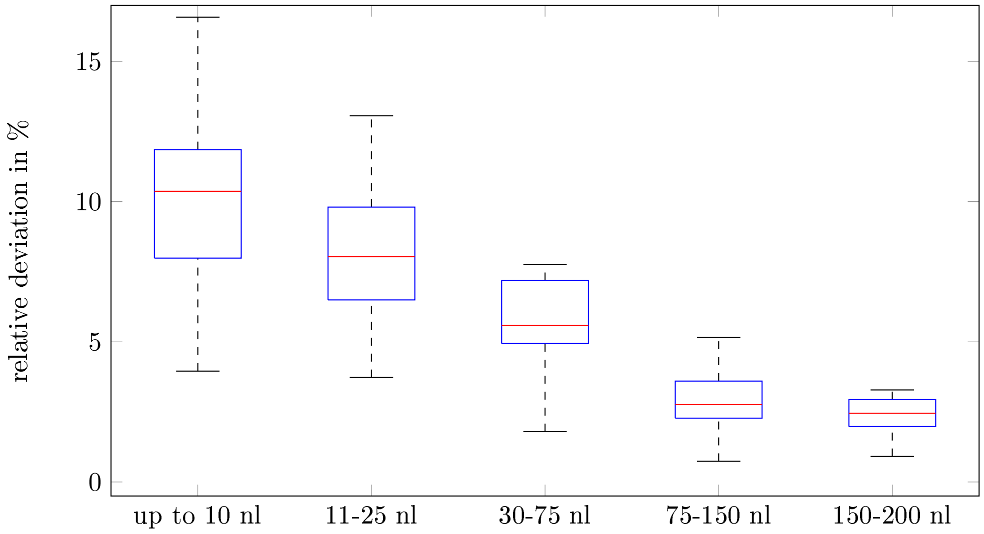


Figure 6.53 Volume Estimation - Repeatability of volumes obtained by 50 measurements of the same PCB. The Deviation is divided by the amount of solder into 5 classes: up to 10 nl, 11-25 nl, 30-75 nl, 75-150 nl, 150-200 nl.

7 Discussion

7.1 General

The overall goal of this thesis was the development and analysis of an online inspection system for PCB jet printing with the capability of being integrated into the PCB jet printer. With respect to these requirements, other inspection methods are not applicable at all or with reasonable effort. The goal was achieved by a normal case photogrammetric approach in the newly defined ultra-close range application field, since a subsequent image acquisition and volume reconstruction is possible during the printing. Therefore, different fields, like calibration and feature processing had to be analyzed and needed further developments to obtain a suitable inspection system. In the following the results of the different fields are discussed.

7.2 Camera Calibration

The proposed pre-processing could detect a sufficient number of calibration markers (at least 30 markers were detected) for the subsequent calibration. The distortion correction results in the proper assignment of point correspondences for the proposed calibration methods. The results showed, that two distortion coefficients can be considered to be sufficient, which coincides with the literature [KB16] [WSZL08] [BK08]. The method does not need any coordinate assignment or correspondence to 2D/3D geometry as it needs to be done e.g. in [TBW03]. However, the geometric layout of the target must be known. The image processing chain and the distortion correction are suitable for other calibration targets (with known geometry) and combinations of spacer rings and lenses. An important fact to mention is the specific design of the lens/spacer combination. An industrial high precision lens with an integrated thread was used to adjust the spacer size without a dismount neither of the camera nor the lens. This ensures additional stability of the imaging model compared to a standard setup where spacer rings are added in between the camera and the lens by unmounting the lens, adding spacers and remounting the lens. As the 2D target was manufactured with high precision, implying an uncertainty of 1 μm and a planar surface, no artifacts caused by a rough surface were visible; thus, the pre-processing steps 2–4 were not required and less neighbors needed to be extracted. Owing to the rougher surface, artifacts may remain in the image of the 3D target. This fact, and the different geometric layouts of the 3D target necessitated 30 neighbors.

It was shown in figures 6.4 and 6.5 that a small operating distance using large spacer rings is important to obtain a small reconstruction error ϵ_{recon}. Owing to this setup, a large magnification in combination with the possible manufacturing uncertainties of a 3D target can result in large deviations in the camera model and a large reprojection error ϵ_{repro_s} using single image calibration. A multi-image calibration, where equal intrinsic camera parameters were used for each acquisition position, eliminates this issue and delivers precise camera models. Thus, these methods, depending on the required accuracy, are the methods of choice for camera calibration in ultra-close range imaging applications. Especially the iterative multi-image calibration yields low reconstruction and reprojection errors similar to the work of Albarelli et. al [ART09]. Instead of a bundle adjustment based reconstruction of a 2D target, the proposed 3D target markers were reconstructed for calibration optimization using a linear triangulation.

The calibration using the model of an optical microscope was not suitable for this work. Similar to the work of Pieters et. al [PJN10] and Zhou et. al [ZN99] the examination plane and sensor plane is parallel and the method results in a significant low reprojection error for all setups between 0.09 and 1 pixel. However, the mentioned studies do not consider any 3D evaluation. The large reconstruction error of more than 1 mm is inappropriate for this work and the method is considered to be unsuitable. Most studies (e.g., [ART09], [FP09], [LRKB13], [LFM16], [BMP^{+}15]) used the reprojection error ϵ_{repro} to verify the calibration results. However, especially in the normal ultra-close range photogrammetry, it was shown (compare figures 6.4 and 6.5) that this value (mean reprojection error ϵ_{repro}) might not be appropriate to conclude on the accuracy of the calibration method. The adapted approach of [ZN99] delivered an excellent reprojection error ϵ_{repro}. However, it was demonstrated that this criterion might not be sufficient for evaluation. In the case of normal case 2D measurements, a low reprojection error ϵ_{repro} will result in proper measurements (distances, areas). However, the corresponding calibrated camera model might result in improper reconstructions in 3D applications. A small reprojection error ϵ_{repro} does not necessarily result in a small reconstruction error ϵ_{recon}. Consequently, in ultra-close range photogrammetry, the 3D reconstruction error ϵ_{recon} should be used.

From tables 6.1 and 6.2 it becomes clear that for the ultra-close range normal case photogrammetry, a large magnification, short operating distance, and wide angle lens are indispensable. This can be achieved using large spacers. Finally, it was shown in section 6.1.4 that the number of distortion parameters exhibits a restricted influence on the resulting camera model. Owing to the results, two parameters are considered as sufficient. Another approach for reconstructing solder joint from images is the bundle adjustment. Due to novel hardware technologies and accelerations bundle adjustment becomes real time capable. However, a direct linear triangulation using calibrated cameras is less computationally intense. Thus, bundle adjustment was not examined in this work.

7.3 Image and 3D Processing

7.3.1 Texture based Solder Paste Detection

The initial pre-processing segments all solder joints but also a high number of non-solder areas. That's why a classification of segmented regions is needed. As shown in the results in section 6.2 the sensitivity as well as the PPV show high values for all data sets, which represents a high precision of the detection/ classification. The high false detection rate in data set 1 is due to the high pollution degree of the PCB surface. Considering every single image the best results are obtained within data set 3. This data set with double-sided illumination allows for a improved acquisition of the solder joints without the loose of information at one side of a solder joint, which is the case for one-sided illuminations, as shown in figure 4.4. Further more data set 3 corresponds to the image quality obtained during manufacturing, since the final acquisition set up includes the double-sided illumination and during the manufacturing only clean boards are used. The positions of non-detected solder joints are at the edge of the image. Due to the ultra-close range normal case setup the camera and its lens respectively has a very small depth of field. Since a wide-angle set up was used with the 10 mm lens, the rays passing through the corners of the image are covering a larger distance than rays through the image center, this distance is slightly outside the depth of field resulting in slight blur effect of these image regions. This blur in turn yields to less structural information of the solder joints and thus to wrong or missing detections. However, when considering the multi-image scanning process a solder joint occurs at multiple positions in the images. If a solder joint is considered to be detected, it is detected in at least one image and transferring the position via the modified camera model to the other images and positions a detection rate of 100% is accessible. The determination of the thresholds for the different Haralick parameters has to be done for every imaging setup. Thus, if illumination changes, a new training set needs to be acquired and classified. However, this is not likely to happen very often since the final imaging setup yields the most efficient data processing and is not considered to be changed. The test PCB contains only structures that are aligned to the PCB edges. That means, due to the normal case imaging setup the solder pads are aligned to the image axes. That leads to bounding boxes that are filled mainly with solder paste. However, in practical applications PCB with rotated solder joints may be manufactured. That would lead to bounding boxes that include a large amount of background pixels, that may disturb the classification. Ferret boxes (bounding box that is aligned to the regions major and minor axis length) can be used to overcome this issue. In that case either the displacement vectors have to be adapted to correspond to the ferret box or the region has to be rotated to be aligned to the image axes.

7.3.2 Feature Detection and Matching

It has been shown in the results, that feature detection and matching for images of solder paste is a non-trivial task. Specific detector settings need to be found, as well as the development of a model based feature correction is indispensable for a proper processing of the input images and to obtain a sufficient point cloud for further post-processing. The described feature detectors have been examined in literature for a wide range of applications (see e.g. [MM12], [CCR^{+}13], [KPS17], [AKMU16] and [TSH^{+}10]. However, the field of feature extraction and matching on images of solder paste was examined for the first time in this work.

Feature Detection A critical parameter for a suitable image acquisition is the camera's integration time. Changing this setting results in a varying image contrast and in a change of structural information of the solder joint areas and thus, prepares the image for a suitable feature detection. Figure 6.17 C shows, that the number of keypoints increases with the integration time and goes into saturation at 18 ms. However, further increasing the integration time does not keep the keypoint distribution. Due to saturated solder joint regions the number of keypoints within these regions decrease, whereas features in the background (PCB surface) occur, since dust particles and scratches become more visible. The novel SolFeC parameter, which is independent of the size of a solder joint (see figure 6.17 B), assesses the feature point coverage of a solder joint area. Evaluating this parameter (figure 6.17 A) results in a suitable camera integration time of 13 ms. This is also congruent to the number of solder joint feature points.
Every considered feature detection configuration results in a different practicability for this application. This is indicated by the computation time and especially by the novel SolFeC parameter. The 5 components of the parameter are necessary to cover different possible feature point densities and distributions. For example, the A_{rel} component can get a value close to one if 4 feature points set up a concave hull that covers most of the considered solder joint, even if no feature points are detected within the solder joint. In that example, the distance between the feature points is much larger than one tenth of the minor axis length. Thus, the d_{le} and w_d components value increases which results in a lower SolFeC parameter. In the opposite case a dense set of feature points may not cover the solder joint properly. In that case, the centroid shift c_{shift} increases and A_{rel} decreases, while d_{le} and w_d deliver suitable values. This also leads to a decrease of the SolFeC parameter as intended. Only a evenly and dense distributed set of feature points lead to a high SolFeC parameter.

Using the MSER detector delivers a SolFeC of 0.3 with a detection time of almost 2 seconds. MSER extracts the centroids of stable regions. A small Δ gray level results in more regions, whereas a large Δ gray level results in a homogeneous region for a solder joint. Independent of this, there can still be a large number of feature points. However, due to the stable region concept the regions are encapsulated resulting in feature points that are very close to each other or lay even above each other. Thus, this feature detector is considered to be not suitable for solder joint feature detection.
One of the most used and well studied feature detectors is SIFT and its accelerated version SURF. They were also examined in this thesis, however, one must note that these detectors are patented, which should be considered if commercial use is intended. SIFT reaches only a poor SolFeC of 0.26. The value becomes lower for a larger contrast threshold, a smaller edge threshold and less octaves. SIFT was developed to detect distinctive image features that are stable along multiple scales where an image is subsampled and smoothed. Due to this processing, the surface structure of solder joints is reduced, resulting in less feature points. Using more octaves can enhance the

number of points since this leads to more stable features. However, this increase is not significant and does not yield a high SolFeC. Allowing for more edge features can enhance the coverage, because small linear features are not treated as edges. The processing time of over 860 ms is also not fast enough for an online capable inspection system.

Similar statements hold for the SURF detector, where processing becomes faster. However, for the best coverage parameter, which is a low value of 0.12, the detector also yields a computation time of 750 ms.

ORB, BRISK and FAST deliver the best SolFeC values. All detectors show a better detection for smaller contrast thresholds. BRISK yields the lowest value with 0.78 and an unusable detection time of over 3.5 seconds per image. ORB and FAST show a saturation of the coverage parameter at a contrast threshold of 10. A further decrease of the contrast threshold results in an increase of non usable features in non-solder regions, which increases the computation time. Since these detectors are computationally less intense, a processing time of 150 ms for ORB and 7 ms for FAST was obtained. Thus, for the further processing ORB and FAST were used. For ORB 5 pyramid levels with a contrast threshold of 10 was selected. The same contrast threshold was set for the FAST feature detection.

Feature Matching The matching of detected image features was performed by applying the feature descriptors. For the best configuration of the FAST and ORB detectors the descriptor matching of ORB, SURF, SIFT, BRISK and BRIEF descriptors and the distance calculation using L1 and L2 norm, hamming distance as well as FLANN were used. In general the FLANN based matching as well as the GPU (CUDA) implementations of the brute force matching yield much faster matchings compared to the brute force CPU matching. Due to the partitioning trees (FLANN) or highly parallel search (CUDA), matches can be found in less than 500 milliseconds. Due to the binary representation of the ORB, BRISK, and BRIEF descriptors the hamming distance (which compares only ones and zeros) yields the fastest matching by using CUDA with less than 300 ms and comparing 2,000,000 descriptors.

The recall and precision curves show the behavior of the matching methods. Mainly precision decreases if the recall rises. That means if more matches are obtained, by increasing the threshold of matching score (vector distance between descriptors) the number of matches increases. However, the increase of false matches is stronger than the increase of correct matches, which results in the lower precision values. When examining the recall precision curves it is noticeable that the decrease of the precision is not universal. Some curves, especially the ORB/BRIEF combination, show a small rise in the precision curve, when increasing the recall. This is due to the homogeneous appearance of solder joints. That means, false matches may establish a higher matching score than correct matches since the descriptors are very similar to each other. The best recall/precision behavior is given by the SIFT and SURF descriptor and the L1 norm. However, these are patented and for a general applicability the third curve which represents the hamming distance of the ORB/BRISK combination is selected for the further processing.

Matching Correction The false matches need to be eliminated to ensure precise point clouds to be reconstructed. Therefore, three different matching corrections were implemented and evaluated. Every correction method was applied independent from each other. The first correction method was performed by the vector distance of descriptors. In general a good match is established if the corresponding descriptors have a small vector distance to each other. False matches can be eliminated by filtering out large vector distances. It has been shown that this statement is not general applicable to solder joint features. Since the descriptors are very similar across different solder joints, a false match can remain and correct matches are removed by setting a vector distance threshold. This is presented in figure 6.41, where false matches, but due to the mentioned similarity, also a large amount of correct matches are removed. This is inappropriate for a subsequent dense feature reconstruction. The matching correction performed by the fundamental matrix was also found to be not suitable for this application. In applications where descriptors differ from each other and where the camera changes its position and rotation, this method yields good corrections. However, in the normal case ultra-close range application of solder joint inspection a lot of features with similar descriptors do remain along the established epipolar line. False matches that differ from this line are perfectly removed, but the false matches along the line remain and result in non sufficient reconstructions. The best correction is given by the camera model based matching correction. For each match, it is checked if the corresponding feature points fit to the adapted camera models. This results in an elimination of matches that do not correspond to the camera models and prepares the matched data set for model based triangulation.

7.3.3 3D Reconstruction, Point Cloud Processing and Volume Estimation

The results have shown, that a sufficient feature matching leads to a proper point cloud that can be post-processed. In this stage of the processing all previous parts (camera calibration, image acquisition and camera model modification, feature detection/matching and correction) came together. A single error in one of the processing steps leads to unusable point clouds. This was shown for a improper matching correction. The further processing aims to a correct volume estimation of single solder joints. Therefore, the clustering was performed to divide the point cloud into solder joint clusters. It was shown that the clustering cannot perform well in terms of separated and solid solder joint clusters if the cloud is not dense enough and well separated at the same time. Due to the sparse nature of the reconstructed point cloud a large cluster distance is needed to encapsulate all points to the same solder joint. However, this enlarged distance ($d > 150\mu m$) yields connected solder joints that are close together on the PCB. This is an unacceptable error and was corrected by the image back projection. The detected solder joints were projected onto the PCB plane using the camera projection matrices to perform a homography.

The projected regions establish a connection between separated 3D points of the same solder joint, whereas maintaining separation in between different solder joints. Thus, the clustering using a small intra-cluster distance results in properly clustered solder joints. The clusters were given to the surface processing which includes the computation of hull vertices and its triangulation with subsequent filtering. The extraction of hull vertices was successful, which was shown by the subsequent triangulation, since the surface triangulation and volume calculation will fail if vertices remain inside the solder joint volume. Inclusion of 3D points would define the triangulation surface as non unique and not closed since each edge must connect only two triangles to define the closed surface. If more triangles are connected to a face the volume calculation will fail which is not the case since all calculated volumes show reasonable values.

From the results in section 6.5 the deviation of the estimated volumes of reconstructed solder joints by the proposed algorithm becomes visible. The estimated volumes of small solder joints show a larger relative deviation. These solder joints are more sensitive to false or missing feature matches due to the smaller amount of feature points, where as the reconstruction of large solder joints is more precise due to the high amount of features and matches. However, the absolute deviation increases with larger solder volumes and becomes more uncertain as it has been shown in figure 6.48 and 6.49. This is due to the required fast online processing. This made necessary to make the acquired PCB regions as large as possible while maintaining solder paste structure and feature point shifts due to solder joint height. Due to the high homogeneous appearance and similar feature descriptors the camera model based correction (including a tolerance region to allow height caused image feature shifting) does not lead to 100% correct matches. Mismatches still occur within the tolerance region. This leads to deviations in the height of reconstructed feature points and thus in the reconstructed surface. Since large solder joint regions contain more matches, more false matches and thus height and volume deviations are possible, resulting in a larger difference to the structured light based volume estimation. In contrast, small solder joints are almost unaffected in terms of absolute deviation. However, the deviation does not rise above 20 nl for large solder joints of 200 nanoliters. Solder joints with a volume larger than 30 nl are measured with a mean deviation of less than 12%. For solder joints with a volume smaller than 25 nl the mean deviation increases up to 16.93%. The overall mean deviation is 14% which is two percent higher compared to the required 12% deviation [Myd10a] described in section 1.3. The repeatability shows no deviation exceeding 12% and thus, is in a practicable range. The precision of the volume estimation can be increased by lowering the working distance and increasing the magnification to allow for more precise feature matching and triangulation including multiple rotated cameras to enhance the intersection of triangulated feature points. However, this would transfer the system from online to offline capability since the processing and acquisition time will increase significantly by more required images to cover all solder joints. Also, additional cameras cannot be attached to be printer head, since the additional weight may influence the motion of the printer head. A statement of the quality of small solder joints can be given with this system without problems.

The values were obtained by comparing the measurements to a reference measurement obtained by a structured light inspection system with an accuracy of 98%. For the examinations test PCBs were employed which contain printable solder joints and other structures. Thus, these PCBs give a good basis for evaluation and demonstration of the successful processing pipeline. However, novel structures or other materials like other solder paste need to be tested and evaluated in future work. Another viscosity of the solder paste may for example result in different structural information in the image, which in turn has to be considered in the solder joint classification and feature detection. Aside to that the provided test PCBs do not contain printing errors. Thus, the behavior of the algorithms could not be evaluated when processing faulty printing results.

8 Conclusion

In this thesis, a novel quality assessment problem in PCB manufacturing that required a specific image acquisition setup, and a precise camera model from calibration in terms of measurements and 3D reconstructions was presented. Herein a photogrammetric image acquisition setup and processing pipeline was developed. The different steps of the photogrammetric process were adapted, implemented and examined. For every part of the processing steps an adequate solution (e.g. for camera calibration, feature matching and surface reconstruction) was found to obtain 3D reconstructions and volume estimations of the printed solder paste.

The proposed multi-image calibration delivered a camera model that yielded a suitable reconstruction error ϵ_{recon}. This small error is crucial for ultra-close range photogrammetry and the corresponding measurements. The acquisition setup that yields a small error (10 mm lens, 5.9 mm spacer, $\epsilon_{recon} = 2.27\ \mu m$) is sufficient for this task.

Feature detection and matching was examined using different detection and matching methods. A new coverage parameter that evaluates the number of keypoints, its distribution and density over a solder joint was introduced. Using the parameters it was found, that popular detectors like SIFT are not suitable in keypointsapplication case. From the 6 examined detectors ORB and FAST delivered the best SolFeC (over 0.95) and a suitable computation time of less than 20 ms. It was shown, that the homogeneous appearance of the solder joints lead to many false matches. State of the art correction methods like vector distance and epipolar geometry based corrections are not able to yield enough correct matches and a low enough number of false matches. Consequently, a matching correction method based on camera models and the inspection geometry which successfully eliminates the false matches was introduced. If a patent-free solution is desired, the FAST detector in combination with a BRIEF descriptor and hamming distance matching with a subsequent camera model based feature matching correction is keypointsmethod of choice. If patented methods can be considered, instead of BRIEF, the SIFT descriptor can be used. For that combination an L2-Norm based matching and the camera model based correction delivers suitable results.

A novel texture based solder joint detection was introduced which is capable to detect 100% of solder joints using overlapping images as it is required by the photogrammetric principle. The detection may be improved by using machine learning in combination with the introduced Haralick parameters. The detected solder joints where back-projected onto the PCB plane for a significant improvement of 3D clustering. Finally a concave hull and Delaunay based surface was created to estimate the solder joints volume from. As it has been shown in the results and the discussion of the volume estimation, the system is not capable of high precision reconstructions and volume measurements, for such a requirement the resolution, magnification and thus feature detection (in terms of accuracy and coverage) is to weak. However, this is not the task of the system. First of all a statement of the presence of solder paste can be given by the texture based solder paste detection stage. The deviation of the volume estimation with respect to the reference method (structured light) exceeds the required 12% for solder joints with a volume smaller than 25 nl. Considering the mean absolute deviation of the volume estimation for these solder joints the value does not exceed 5 nl, which coincides with the smallest printable solder joint, and does not exceed 8 nl in the mean value at all. If this is sufficient for a quality statement needs to be tested in a practical use case. 55% of the examined solder joints show a deviation of less than 12%. The deviation of the estimated volume increases with the size of the solder joint. For larger joints the deviation stays under the required 12%. The mean deviation over all measured solder joints has a value of 14%. Improvements to this, in a photogrammetric sense, can be done by a larger magnification and the use of rotated camera systems and telecentric lenses. However, this would lead the system towards a stand-alone solution, where smaller areas of the PCBs are captured. The usage of GPU-based accelerated image processing and the implemented pipeline design using Intel TBB results in an online capability of the system, where multiple steps of the process chain work in parallel for each image. Due to the modular software and hardware development the system is able to work as a stand alone inspection system as well as an integrated system and can be extended easily. For an integration into the printer future work needs access to the control data and interfaces of the printer as well as the input PCB design (gerber) data to perform an built-in inspection. This data can also be used to achieve a further speed up by the usage of this design data. The known position of solder pads can be used as regions of interest for solder joint detection and thus saves time during detection. However, if smudged solder joints should be detected, the presented pipeline, especially the texture based solder joint detection, is indispensable. Developments in sensor, interface and lens technology can improve the acquisition and reconstruction process, for example if a high resolution sensor ($>$ 4MP) acquires images at high speed and high image quality in terms of noise and sharpness, a more detailed and more precise reconstruction can be obtained. However, this increase of data leads to a higher computation times in process steps like feature detection/matching and clustering. This could be solved by using multi-GPU systems or FPGA systems by implementing extensively parallelized algorithms and processing chains. A further speedup of the whole processing could be obtained by determining if a solder joint is covered by enough images and features and starting the 3D processing pipeline in another process parallel to the image acquisition and reconstruction pipeline.

Statutory Declaration

I declare, that I have authored this thesis independently, that I have not used other than the declared sources / resources, and that I have explicitly marked all material which has been quoted either literally or by content from the used sources. Images and drawings without any references were made by myself or come are produced by the developed software.

Erklärung

Ich versichere, dass ich die vorliegende Arbeit ohne unzulässige Hilfe Dritter und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe. Die aus anderen Quellen direkt oder indirekt übernommenen Daten und Konzepte sind unter Angabe der Quelle gekennzeichnet. Andere Personen waren an der inhaltlich-materiellen Erstellung der vorliegenden Arbeit nicht beteiligt. Insbesondere habe ich hierfür nicht die entgeltliche Hilfe von Vermittlungs- bzw. Beratungsdiensten (Promotionsberater oder anderer Personen) in Anspruch genommen. Niemand hat von mir unmittelbar oder mittelbar geldwerte Leistungen für Arbeiten erhalten, die im Zusammenhang mit dem Inhalt der vorgelegten Dissertation stehen. Die Arbeit wurde bisher weder im In- noch im Ausland in gleicher oder ähnlicher Form einer Prüfungsbehörde vorgelegt. Ich bin darauf hingewiesen worden, dass die Unrichtigkeit der vorstehenden Erklärung als Täuschungsversuch bewertet wird und gemäß § 7 Abs. 10 der Promotionsordnung den Abbruch des Promotionsverfahrens zur Folge hat.

Ilmenau, 15. November 2019

(Ort, Datum)	(David Heinemann)

References

[AKMU16] E Ali, Engr Sajid Ullah Khan, Engr Muhammad Zarrar Mahmudi, and Rahmat Ullah. A Comparison of FAST, SURF, Eigen, Harris, and MSER Features. *Int. J. Comput. Eng. Inf. Technol*, 8(6):100–105, 2016.

[ALDF94] Abdalmajeid M. Alyassin, Jack L. Lancaster, J. Hunter Downs, and Peter T. Fox. Evaluation of new algorithms for the interactive measurement of surface area and volume. *Medical Physics*, 21(6):741–752, jun 1994.

[ART09] Andrea Albarelli, Emanuele Rodolà, and Andrea Torsello. Robust Camera Calibration using Inaccurate Targets. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31:376–383, 2009.

[Bau12] Baumer Optronic GmbH. *Baumer HXC SeriesUser's Guide for CameraLink Cameras with CMOSIS Sensors*. Radeberg, 2012.

[BB15] Wilhelm Burger and Mark James Burge. *Digitale Bildverarbeitung - Eine algorithmische Einführung mit Java, 3. Auflage*. Springer, 2015.

[BETG08] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool. Speeded-Up Robust Features (SURF). *Computer Vision and Image Understanding*, 110(3):346–359, 2008.

[BFH01] Ján Bartl, Roman Fíra, and Miroslav Hain. Inspection of surface by the moiré method. *Measurement Science Review*, 1(1):29–32, 2001.

[Bie11] Henk Biemans. 5D solder paste inspection - merits beyond 3D technology. *Global SMT & Packaging*, 2011:8–13, 2011.

[BK87] K. L. Boyer and A. C. Kak. Color-Encoded Structured Light for Rapid Active Ranging. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-9(1):14–28, 1987.

[BK08] Gary Bradski and Adrian Kaehler. *Learning OpenCV*. O'Reilly Media, Inc., Sebastopol, CA, USA, 2008.

[BKJJ13] Csaba Benedek, Olivér Krammer, Mihály Janoczki, and László Jakab. Solder Paste Scooping Detection by Multilevel Visual Inspection of Printed Circuit Boards. *IEEE Transactions on Industrial Electronics*, 60(6):2318–2331, 2013.

[BMP$^+$15] Jan Behmann, Anne-Katrin Mahlein, Stefan Paulus, Heiner Kuhlmann, Erich-Christian Oerke, and Lutz Plümer. Calibration of hyperspectral close-range pushbroom cameras for plant phenotyping. *ISPRS Journal of Photogrammetry and Remote Sensing*, 106:172–182, 2015.

[Bro66] Duane C. Brown. Decentering Distortion of Lenses. *Photometric Engineering, Vol. 32, No. 3*, pages 444 – 462, 1966.

[BTVG06] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. Surf: Speeded up robust features. In *European conference on computer vision*, pages 404–417. Springer, 2006.

[CCR+13] Antonio Canclini, Matteo Cesana, Alessandro Redondi, Marco Tagliasacchi, João Ascenso, and R Cilla. Evaluation of low-complexity visual feature detectors and descriptors. In *International Conference on Digital Signal Processing (DSP)*, pages 1–7. IEEE, 2013.

[CK98] Yi-Bae Choi and Seung-Woo Kim. Phase-shifting grating projection moire topography. *Optical Engineering*, 37(3):1005 – 1010, 1998.

[CLL+88] H. E. Cline, W. E. Lorensen, S. Ludke, C. R. Crawford, and B. C. Teeter. Two algorithms for the three-dimensional reconstruction of tomograms. *Medical Physics*, 15(3):320–327, may 1988.

[CLSF10] Michael Calonder, Vincent Lepetit, Christoph Strecha, and Pascal Fua. BRIEF: Binary Robust Independent Elementary Features. In *Computer Vision – ECCV 2010*, pages 778–792, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg.

[Coe2] Nico Coenen. Lotpastenauftrag mit JetPrinting Technologie ermöglicht neue Möglichkeiten in der SMT-Fertigung. `http://www.etfn.de/uploads/media/Mydata.pdf`, 2. Accessed: 2018-08-01.

[dBCvKO08] Mark de Berg, Otfried Cheong, Marc van Kreveld, and Mark Overmars. *Computational Geometry*. Springer Berlin Heidelberg, 2008.

[Dua71] C. Brown Duane. Close range camera calibration. *Photogrammetric Engineering*, 37(8):855–866, 1971.

[DZ02] X. Descombes and J. Zerubia. Marked point process in image analysis. *IEEE Signal Processing Magazine*, 19(5):77–84, 2002.

[EM94] Herbert Edelsbrunner and Ernst P. Mücke. Three-dimensional alpha shapes. *ACM Transactions on Graphics*, 13(1):43–72, 1994.

[FBF77] Jerome H. Friedman, Jon Louis Bentley, and Raphael Ari Finkel. An Algorithm for Finding Best Matches in Logarithmic Expected Time. *ACM Transactions on Mathematical Software*, 3(3):209–226, 1977.

[FP09] Yasutaka Furukawa and Jean Ponce. Accurate camera calibration from multi-view stereo and bundle adjustment. *International Journal of Computer Vision*, 84(3):257–268, 2009.

[FPF99] Andrew Fitzgibbon, Maurizio Pilu, and Robert B Fisher. Direct least square fitting of ellipses. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(5):476–480, 1999.

[FW16] Wolfgang Förstner and Bernhard P. Wrobel. *Photogrammetric Computer Vision*. Springer International Publishing, Basel, 2016.

[GL84] Konecny Gottfried and Gerhard Lehmann. *Photogrammetrie*. Berlin/New York: Walter de Gruyter, 1984.

[GM04] Sing Bing Kang Gerard Medioni. *Emerging Topics in Computer Vision*. PRENTICE HALL, 2004.

[Gra16] Stuart I Granshaw. Photogrammetric terminology. *The Photogrammetric Record*, 31(154):210–252, 2016.

[Han09] Heinz Handels. *Medizinische Bildverarbeitung*. Vieweg und Teubner, 2009.

[HLYK11] Chien-Yi Huang, Yueh-Hsun Lin, Kuo-Ching Ying, and Chen-Liang Ku. The solder paste printing process: critical parameters, defect scenarios, specifications, and cost reduction. *Soldering & Surface Mount Technology*, 23(4):211–223, 2011.

[Hor00] Berthold K. P. Horn. Tsai's camera calibration method revisited. `http://people.csail.mit.edu/bkph/articles/Tsai_Revisited.pdf`, 2000. Accessed: 14.11.2019.

[HP08] Tak-Wai Hui and Grantham Kwok-Hung Pang. 3-D Measurement of Solder Paste Using Two-Step Phase Shift Profilometry. *IEEE Transactions on Electronics Packaging Manufacturing*, 31(4):306–315, 2008.

[HS88] Chris Harris and Mike Stephens. A combined corner and edge detector. In *In Proc. of Fourth Alvey Vision Conference*, pages 147–151, 1988.

[Hug10] Wolfgang Hugemann. Correcting Lens Distortions in Digital Photographs. In *19. EVU Conference*, 2010.

[HvACL96] Jef L. Horijon, Willem D. van Amstel, Fred C. M. Couweleers, and Wilco C.A. Ligthart. Optical system of an industrial 3D laser scanner for solder paste inspection. In Michael R. Descour, Kevin G. Harding, and Donald J. Svetkoff, editors, *Three-Dimensional and Unconventional Imaging for Industrial Inspection and Metrology*, volume 2599, pages 162 – 170. International Society for Optics and Photonics, SPIE, 1996.

[HZ03] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Pr., 2003.

[Int18] ASC International. Yield Improvement Strategy. `http://www.ascinternational.com/index.php/content/yield_improvement_strategy`, June 2018. Accessed: 2018-08-01.

[Jä12] Bernd Jähne. *Digitale Bildverarbeitung*. Springer Berlin Heidelberg, 2012.

[JCT12] Jun Jiang, Jun Cheng, and Dacheng Tao. Color Biological Features-Based Solder Paste Defects Detection and Classification on Printed Circuit Boards. *IEEE Transactions on Components, Packaging and Manufacturing Technology*, 2(9):1536–1544, 2012.

[JS10] Edward Kandrot Jason Sanders. *Cuda by Example: An Introduction to General-Purpose Gpu Programming*. ADDISON WESLEY PUB CO INC, 2010.

[KB16] Adrian Kaehler and Gary Bradski. *Learning OpenCV 3: Computer Vision in C++ with the OpenCV Library*. O'Reilly Media, Inc., Sebastopol, 2016.

[KBH06] Michael Kazhdan, Matthew Bolitho, and Hugues Hoppe. Poisson surface reconstruction. In *Proceedings of the Fourth Eurographics Symposium on Geometry Processing*, SGP '06, pages 61–70, Aire-la-Ville, Switzerland, Switzerland, 2006. Eurographics Association.

[KCMP99] Tae-Hyeon Kim, Tai-Hoon Cho, Young Shik Moon, and Sung Han Park. Visual inspection system for the classification of solder joints. *Pattern Recognition*, 32(4):565–575, 1999.

[KCO99] Seung-Woo Kim, Yi-Bae Choi, and Jung-Taek Oh. Reverse engineering: high speed digitization of free-form surfaces by phase-shifting grating projection moiré topography. *International Journal of Machine Tools and Manufacture*, 39(3):389–401, mar 1999.

[KL80] Virginia C. Klema and Alan J. Laub. The Singular Value Decomposition: Its Computation and Some Applications. *IEEE Transactions on Automatic Control,*, 25(2):164–176, 1980.

[KM88] Catherine A. Keely and Charles C. Morehouse. Structured Light Technique Applied To Solder Paste Height Measurement. In Donald J. Svetkoff, editor, *Optics, Illumination, and Image Sensing for Machine Vision II*, volume 0850, pages 88 – 92. International Society for Optics and Photonics, SPIE, 1988.

[Kon08] F. Kong. A new method of inspection based on shape from shading. In *Congress on Image and Signal Processing*, volume 2, pages 291–294, 2008.

[KPS17] Ebrahim Karami, Siva Prasad, and Mohamed S. Shehata. Image matching using sift, surf, BRIEF and ORB: performance comparison for distorted images. *CoRR*, abs/1710.02726, 2017.

[KYWY06] Chung-Hsien Kuo, Fang-Chung Yang, Jein-Jong Wing, and Ching-Kun Yang. Construction of 3D solder paste surfaces using multi-projection images. *The International Journal of Advanced Manufacturing Technology*, 31(5-6):509–519, 2006.

[LBKP16] Jinjin Li, Bonnie L. Bennett, Lina J. Karam, and Jeff S. Pettinato. Stereo Vision Based Automated Solder Ball Height and Substrate Coplanarity Inspection. *IEEE Transactions on Automation Science and Engineering*, 13(2):757–771, 2016.

[LCS11] Stefan Leutenegger, Margarita Chli, and Roland Y Siegwart. BRISK: Binary robust invariant scalable keypoints. In *IEEE International Conference on Computer Vision (ICCV)*, pages 2548–2555. IEEE, 2011.

[Lee83] Jong-Sen Lee. Digital image smoothing and the sigma filter. *Computer Vision, Graphics, and Image Processing*, 24(2):255–269, 1983.

[Lev44] Kenneth Levenberg. A Method for the Solution of certain non-linear Problems in least Squares. *Quarterly of Applied Mathematics*, 2(2):164–168, 1944.

[LFM16] Thomas Luhmann, Clive Fraser, and Hans-Gerd Maas. Sensor modelling and camera calibration for close-range photogrammetry. *ISPRS Journal of Photogrammetry and Remote Sensing*, 115:37–46, 2016.

[LL99] Horng-Hai Loh and Ming-Sing Lu. Printed circuit board inspection using image analysis. *IEEE Transactions on Industry Applications*, 35(2):426–432, 1999.

[Low99] David G Lowe. Object recognition from local scale-invariant features. In *The proceedings of the seventh IEEE international conference on Computer vision 1999.*, volume 2, pages 1150–1157, 1999.

[Low04] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.

[LRKB13] Thomas Luhmann, Stuart Robson, Stephen Kyle, and Jan Boehm. *Close-Range Photogrammetry and 3D Imaging.* Walter de Gruyter GmbH, Berlin, 2013.

[LRKH11] Thomas Luhman, Stuart Robson, Stephen Kyle, and Ian Harley. *Close Range Photogrammetry: Principles, Techniques and Applications.* Whittles Publishing, Dunbeath, 2011.

[LSZD07] Bin Liu, Changku Sun, Xiaodong Zhang, and Xiaobing Deng. 3d laser measurement technique of solder paste. In Liwei Zhou, editor, *International Symposium on Photoelectronic Detection and Imaging 2007: Image Processing*. SPIE, sep 2007.

[Luh14] Thomas Luhmann. Eccentricity in images of circular and spherical targets and its impact to 3D object reconstruction. *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 40(5):363, 2014.

[Mar63] Donald J. Marquardt. An Algorithm for Least-Squares Estimation of Nonlinear Parameters. *Journal of the Society for Industrial and Applied Mathematics*, 11(2):431–441, 1963.

[Mat08] Neffra A Matthews. *Aerial and close-range photogrammetric technology: providing resource documentation, interpretation, and preservation.* US Department of the Interior, Bureau of Land Management, 2008.

[MCUP04] J Matas, O Chum, M Urban, and T Pajdla. Robust wide-baseline stereo from maximally stable extremal regions. *Image and Vision Computing*, 22(10):761 – 767, 2004. British Machine Vision Computing 2002.

[ML09] Marius Muja and David G. Lowe. Fast Approximate Nearest Neighbors with Automatic Algorithm Configuration. In *International Conference on Computer Vision Theory and Application VISSAPP'09)*, pages 331–340. INSTICC Press, 2009.

[ML12] Marius Muja and David G. Lowe. Fast Matching of Binary Features. In *Computer and Robot Vision (CRV)*, pages 404–410, 2012.

[ML14] Marius Muja and David G. Lowe. Scalable Nearest Neighbor Algorithms for High Dimensional Data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36, 2014.

[MM12] Ondrej Miksik and Krystian Mikolajczyk. Evaluation of local detectors and descriptors for fast feature matching. In *Pattern Recognition (ICPR), 2012 21st International Conference on*, pages 2681–2684. IEEE, 2012.

[MM15] Satyajit Mondal and Joydeep Mukherjee. Image Similarity Measurement using Region Props, Color and Texture: An Approach. *International Journal of Computer Applications*, 121(22), 2015.

[MMB04] C McGlone, E Mikhail, and J Bethel. Manual of Photogrammetry, American Society for Photogrammetry and Remote Sensing. *Bethesda, MD*, 2004.

[MYD07] MYDATA automation AB. MY500 - Jet Printing Technology. `http://support.mydata.com/MYWEB/Mult.nsf/lupgraphics/P_001_0181_MY500ProductFlyer.`

`pdf/$File/P_001_0181_MY500ProductFlyer.pdf`, June 2007. Accessed: 14.11.2019.

[Myd08] Mydata. *My500 JetPrinter Service Manual.* Mydata automation AB, Adolfsbergsvägen 11 16866 Bromma, June 2008.

[Myd10a] Mydata. My500 Jet Printer Specification. `http://isk-smt.de/fileadmin/user_upload/MY500Specification04_2010.pdf`, Täby, April 2010. Accessed: 2019-11-13.

[Myd10b] Mydata. We've joined the jet printing movement. Let us tell you why. `https://docplayer.net/48813742-We-ve-joined-the-jet-printing-movement-let-us-tell-you-why.html`, Täby, January 2010. Accessed: 2019-11-13.

[NS14] Takeshi Nakazawa and Ayman Samara. Three-dimensional inline inspection for substrate warpage and ball grid array coplanarity using stereo vision. *Applied Optics*, 53(14):3101, 2014.

[OKO$^+$95] T. Okura, M. Kanai, S. Ogata, T. Takei, and H. Takakusagi. Optimization of solder paste printability with laser inspection technique. In *Seventeenth IEEE/CPMT International Electronics Manufacturing Technology Symposium. 'Manufacturing Technologies - Present and Future'*, pages 361–365. IEEE, 1995.

[OMR] OMRON. 3D-SJI: 3D Solder joint inspection. `https://inspection.omron.eu/en/3d-sji`. Accessed: 2018-08-01.

[Ots79] Nobuyuki Otsu. A Threshold Selection Method from Gray-Level Histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, 9(1):62–66, 1979.

[PC09] Grantham K.H. Pang and Ming-Hei Chu. Automated optical inspection of solder paste based on 2.5D visual images. In *2009 International Conference on Mechatronics and Automation*, pages 982–987. IEEE, 2009.

[PJN10] R. S. Pieters, P .P. Jonker, and H. Nijmeijer. Product Pattern-Based Camera Calibration for Microrobotics. *25th International Conference of Image and Vision Computing New Zealand*, 2010.

[PTS$^+$04] J. Pan, G. L. Tonkay, R. H. Storer, R. M. Sallade, and D. J. Leandri. Critical Variables of Solder Paste Stencil Printing for Micro-BGA and Fine-Pitch QFP. *IEEE Transactions on Electronics Packaging Manufacturing*, 27(2):125–132, 2004.

[Ran04] Ananth Ranganathan. The Levenberg-Marquardt Algorithm. `https://pdfs.semanticscholar.org/21a4/55a2847f11f41ccf7b90269c7e5670450eb0.pdf`, 2004. Accessed: 14.11.2019.

[RD06] Edward Rosten and Tom Drummond. Machine Learning for High-Speed Corner Detection. In Aleš Leonardis, Horst Bischof, and Axel Pinz, editors, *Computer Vision – ECCV 2006*, pages 430–443, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.

[Rei07] James Reinders. *Intel Threading Building Blocks: Outfitting C++ for Multi-core Processor Parallelism.* O'Reilly Media, 2007.

[RRKB11] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. ORB: An efficient alternative to SIFT or SURF. In *Computer Vision (ICCV), 2011 IEEE international conference on*, pages 2564–2571. IEEE, 2011.

[RUS09] Radu Bogdan RUSU. Semantic 3D Object Maps for Everyday Manipulation in Human Living Environments. 260 p. *Disertation thesis*, 2009.

[RVSS10] Carlos Ricolfe-Viala and Antonio-José Sánchez-Salmerón. Lens distortion models evaluation. *Applied Optics*, Vol. 49(No. 30):5914 – 5928, 2010.

[SC01] Xianyu Su and Wenjing Chen. Fourier transform profilometry: A review. *Optics and Lasers in Engineering*, 35:263–284, 05 2001.

[Sco76] P. J. Scott. Close Range Camera Calibratin: A new Method. *Photogrammetric Record*, 8(48):806–812, October 1976.

[SF16] Johannes L. Schonberger and Jan-Michael Frahm. Structure-from-Motion Revisited. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR).* IEEE, 2016.

[SL12] Yaqi Sun and Yu Liu. Research on SMT solder joint image segmentation. In *2012 13th International Conference on Electronic Packaging Technology & High Density Packaging*, pages 1287–1289. IEEE, 2012.

[SML06] Will Schroeder, Ken Martin, and Bill Lorensen. *Visualization Toolkit: An Object-Oriented Approach to 3D Graphics, 4th Edition.* Kitware, 2006.

[SPB04] Joaquim Salvi, Jordi Pagès, and Joan Batlle. Pattern codification strategies in structured light systems. *Pattern Recognition*, 37(4):827–849, 2004.

[TBW03] Thorsten Thormaehlen, Hellward Broszio, and Ingolf Wassermann. Robust Line-Based Calibration of Lens Distortion from a Single View. In *Proc*, pages 105–112, 2003.

[TM83] Mitsuo Takeda and Kazuhiro Mutoh. Fourier transform profilometry for the automatic measurement of 3-D object shapes. *Applied Optics*, 22(24):3977, 1983.

[TMHF00] Bill Triggs, Philip F. McLauchlan, Richard I. Hartley, and Andrew W. Fitzgibbon. Bundle Adjustment — A Modern Synthesis. In Bill Triggs, Andrew Zisserman, and Richard Szeliski, editors, *Vision Algorithms: Theory and Practice*, pages 298–372, Berlin, Heidelberg, 2000. Springer Berlin Heidelberg.

[TMTF07] Atsushi Teramoto, Takayuki Murakoshi, Masatoshi Tsuzaka, and Hiroshi Fujita. Automated Solder Inspection Technique for BGA-Mounted Substrates by Means of Oblique Computed Tomography. *IEEE Transactions on Electronics Packaging Manufacturing*, 30(4):285–292, 2007.

[Toe12] Klaus D. Toennies. *Guide to Medical Image Analysis*. Springer London, 2012.

[Tsa87] Roger Y. Tsai. A Versatile Camera Calibration Technique for High-Accuracy 3D Machine Vision Metrology Using Off-the-shelf TV Cameras and Lenses. *IEEE Journal of Robotics and Automation, Vol. RA-3, No. 4*, 3(4):323–344, 1987.

[TSH$^+$10] Yuehua Tao, Marjorie Skubic, Tony Han, Youming Xia, and Xiaoxiao Chi. Performance Evaluation of SIFT-Based Descriptors for Object Recognition. *Proceedings of the International MultiConference of Engineers and Computer Scientists*, 2:1453–1456, 2010.

[UMG12] Rubén Usamentiaga, Julio Molleda, and Daniel F. García. Fast and robust laser stripe extraction for 3D reconstruction in industrial environments. *Machine Vision and Applications*, 23(1):179–196, 2012.

[Wat81] David F. Watson. Computing the n-dimensional Delaunay tessellation with application to Voronoi polytopes. *The computer journal*, 24(2):167–172, 1981.

[WSZL08] Jianhua Wang, Fanhuai Shi, Jing Zhang, and Yuncai Liu. A new calibration model of camera lens distortion. *Pattern Recognition*, 41(2):607–615, 2008.

[WZKL08] Huihui Wu, Xianmin Zhang, Yongcong Kuang, and Shenglin Lu. A real-time machine vision system for solder paste inspection. In *2008 IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, pages 205–210. IEEE, 2008.

[Zha99] Zhengyou Zhang. Flexible camera calibration by viewing a plane from unknown orientations. In *The Proceedings of the Seventh IEEE International Conference on Computer Vision*, volume 1, pages 666–673, 1999.

[Zha00] Z. Zhang. A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(11):1330–1334, 2000.

[Zha04] Zhengyou Zhang. Camera calibration with one-dimensional objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(7):892–899, 2004.

[ZN99] Yu Zhou and Bradley J. Nelson. Calibration of a parametric model of optical microscope. *Optical Engineering*, 38(12):1989–1995, 1999.

A Appendix

A.1 Calibration Target Engineering Drawing

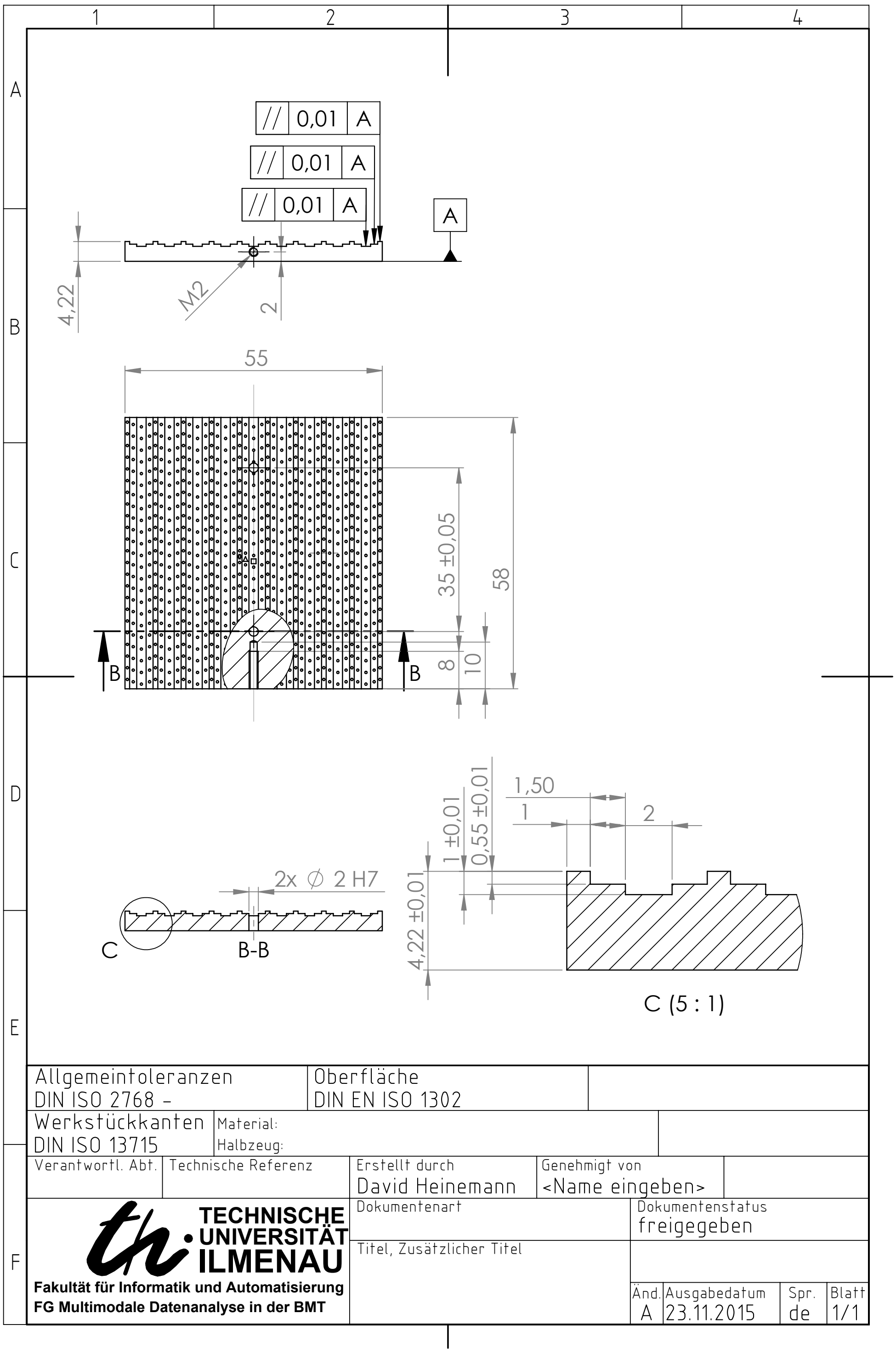
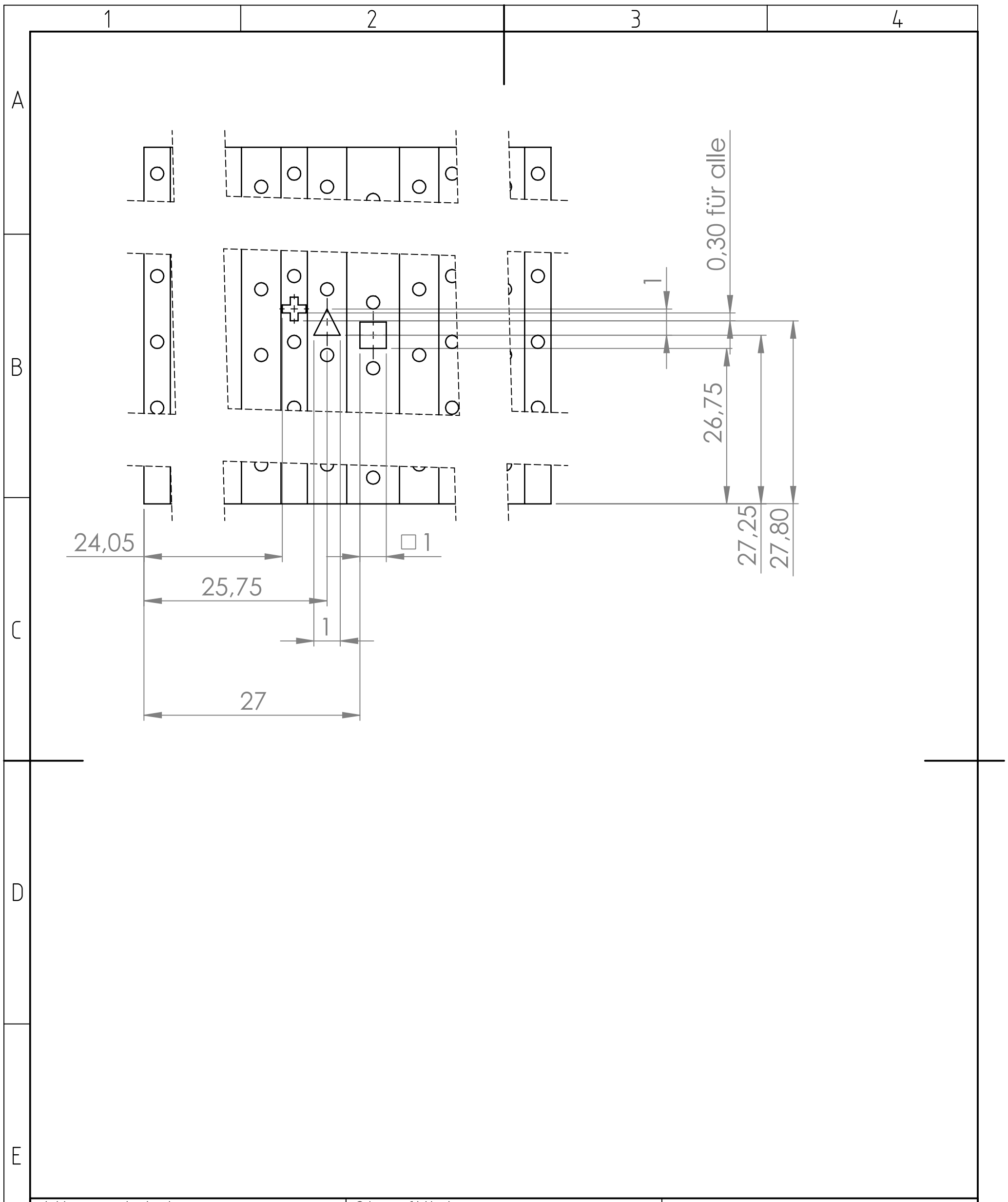


Figure A.1 Engineering drawing of 3D calibration target.



Allgemeintoleranzen
DIN ISO 2768 -
Oberfläche
DIN EN ISO 1302
Werkstückkanten
DIN ISO 13715
Material:
Halbzeug:
Verantwortl. Abt.
Technische Referenz
Erstellt durch
David Heinemann
Genehmigt von
<Name eingeben>
TECHNISCHE UNIVERSITÄT ILMENAU
Fakultät für Informatik und Automatisierung
FG Multimodale Datenanalyse in der BMT
Dokumentenart
Dokumentenstatus
freigegeben
Titel, Zusätzlicher Titel

Änd.	Ausgabedatum	Spr.	Blatt
A	23.11.2015	de	1/1

Figure A.2 Engineering drawing of 3D calibration target - Detail 1.

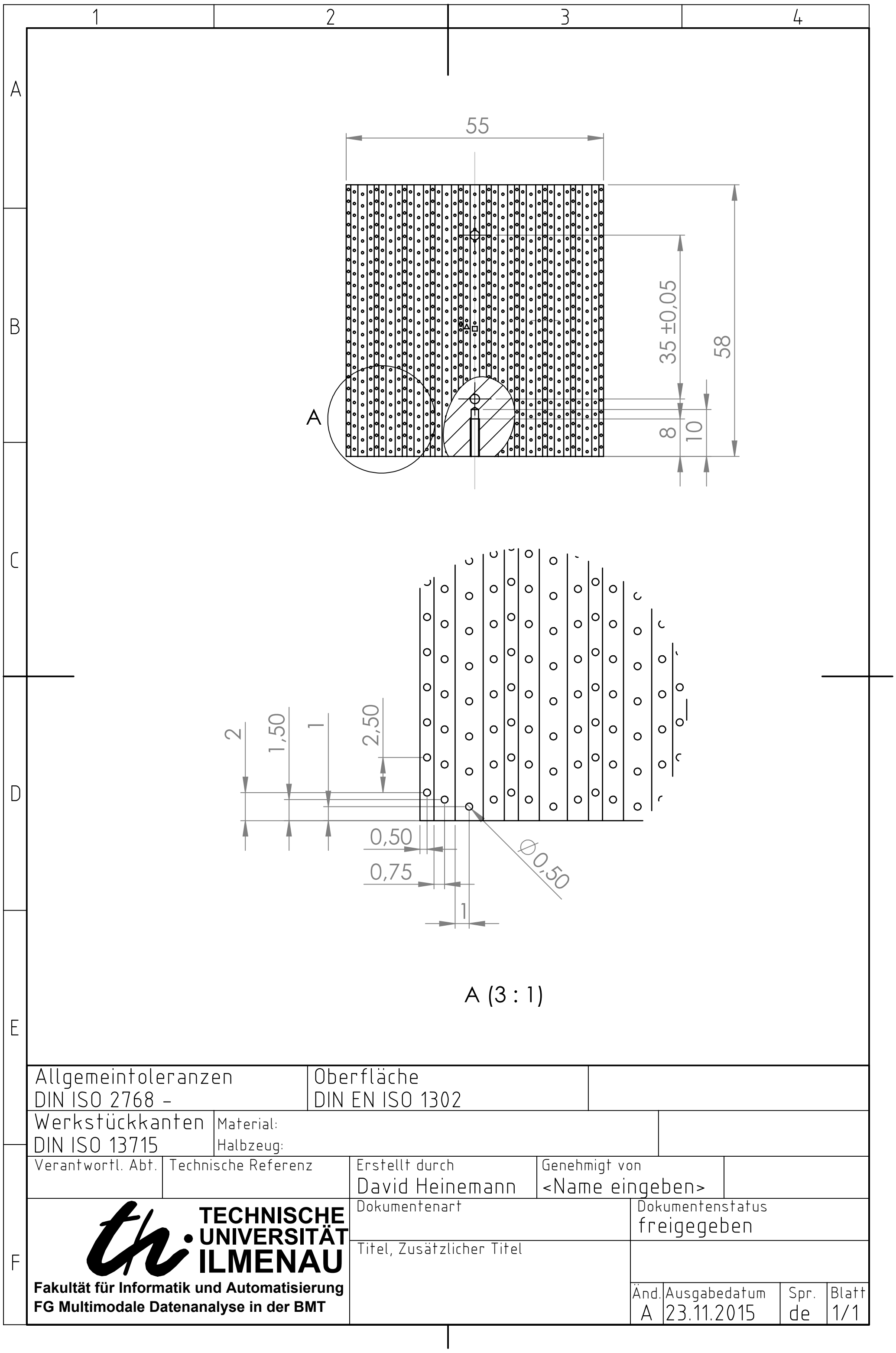


Figure A.3 Engineering drawing of 3D calibration target - Detail 2.

A.2 Haralick Texture Features

Parameters

Contrast:

$$CON_\delta = \sum_{i=0}^{g-1}\sum_{j=0}^{g-1}(i-j)^2 P_\delta(i,j) \tag{A.1}$$

Correlation:

$$COR_\delta = \frac{\sum_i^{g-1}\sum_j^{g-1}(ijP_\delta(i,j) - \mu_{\delta_x}\mu_{\delta_y})}{\sigma_{\delta_x}\sigma_{\delta_y}} \tag{A.2}$$

Difference Variance:

$$DV_\delta = \sum_{i=0}^{g-1}(i-\mu_{\delta_{x-y}})^2 P_{\delta_{x-y}}(i) \tag{A.3}$$

Difference Entropy:

$$DE_\delta = -\sum_{i=0}^{g-1} P_{\delta_{x-y}}(i) log(P_{\delta_{x-y}}(i)) \tag{A.4}$$

Entropy:

$$ENT_\delta = -\sum_{i=0}^{g-1}\sum_{j=0}^{g-1} P_\delta(i,j) log(P_\delta(i,j)) := H_{x,y} \tag{A.5}$$

Inverse Difference Moment (Homogeneity):

$$IDM_\delta = \sum_{i=0}^{g-1}\sum_{j=0}^{g-1} \frac{P_\delta(i,j)}{1+(i-j)^2} \tag{A.6}$$

Information Measure of Correlation I:

$$(H_{\delta_{x,y}} - H^1_{\delta_{x,y}})/\{H_{\delta_x}, H_{\delta_y}\} \tag{A.7}$$

Information Measure of Correlation II:

$$\sqrt{1 - exp(-2(H^2_{\delta_x} - H_{\delta_y}))} \tag{A.8}$$

Maximal Correlation Coefficient: square root of the second largest Eigenvalue of Q with:

$$Q(i,j) = \sum_k \frac{P_\delta(i,k)P_\delta(j,k)}{P_{\delta_x}(i)P_{\delta_y}(k)} \tag{A.9}$$

Second Angular Moment (Energy):

$$SAM_\delta = \sum_{i=0}^{g-1}\sum_{j=0}^{g-1}(P_\delta(i,j))^2 \tag{A.10}$$

Sum Average:

$$SA_\delta = \sum_{i=2}^{2g} i P_{\delta_{x+y}}(i) \tag{A.11}$$

Sum Entropy:

$$SE_\delta = -\sum_{i=2}^{2g} P_{\delta_{x+y}}(i) log(P_{\delta_{x+y}}(i)) \tag{A.12}$$

Sum Variance:

$$SV_\delta = \sum_{i=2}^{2g} (i - SE_\delta)^2 P_{x+y}(i) \tag{A.13}$$

Variance:

$$VAR_\delta = \sum_{i=0}^{g-1} \sum_{j=0}^{g-1} (i - \mu)^2 P_\delta(i,j) \tag{A.14}$$

Results of Haralick Parameters in Solder and non-Solder Regions

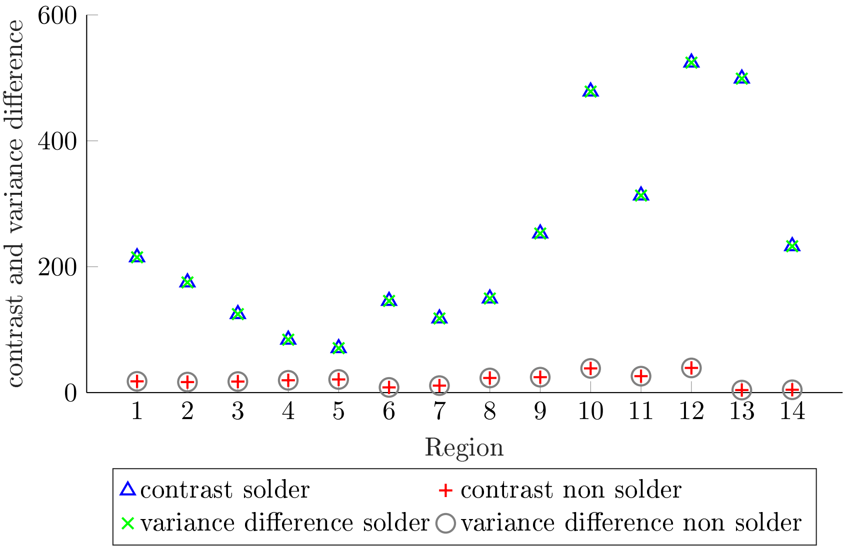


Figure A.4 Comparison of contrast and variance difference of solder paste (blue) and non-solder structures (red). Regions were randomly selected. The variance difference values do not differ from the contrast significantly. Thus, its usage does not enhance the decision precision.

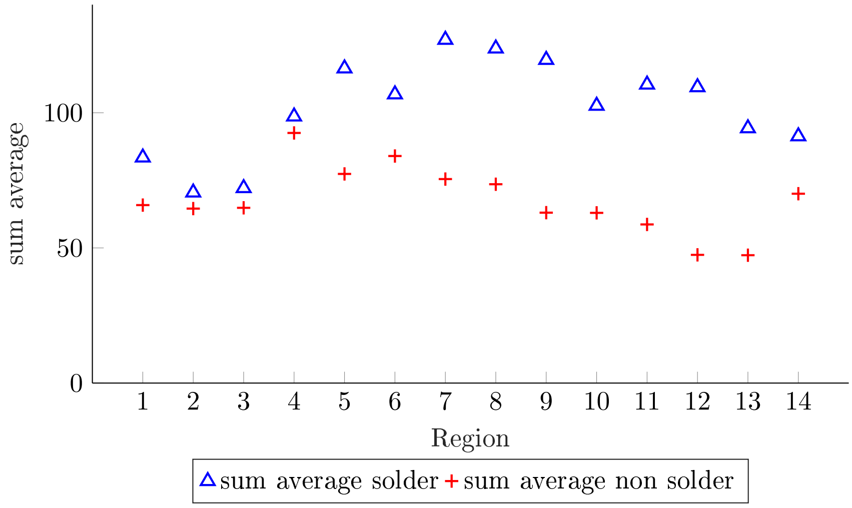


Figure A.5 Comparison of sum average values of solder paste (blue) and non-solder structures (red). Regions were randomly selected. The sum average of solder joints overlaps the contrast of non-solder structures, resulting in an impossible separation of the data sets.

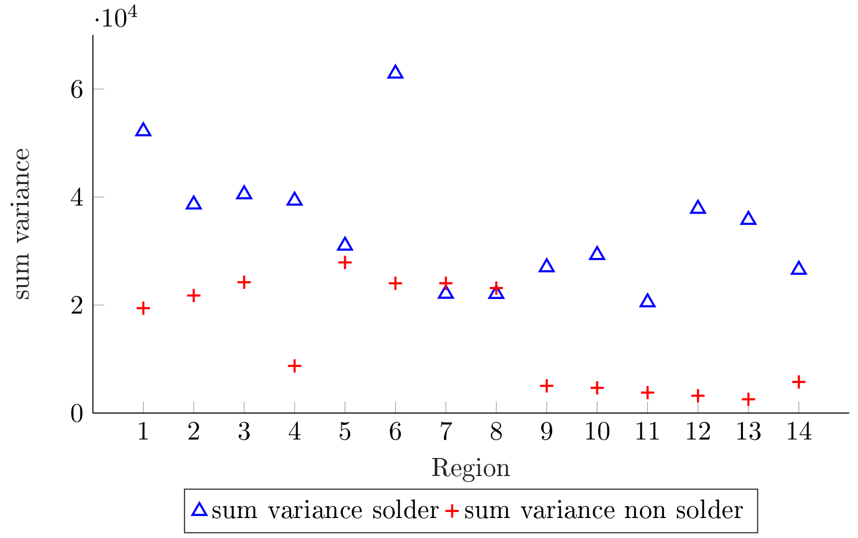


Figure A.6 Comparison of contrast values of solder paste (blue) and non-solder structures (red). Regions were randomly selected. The sum variance of solder joints cannot be separated from the sum variance of non-solder areas.

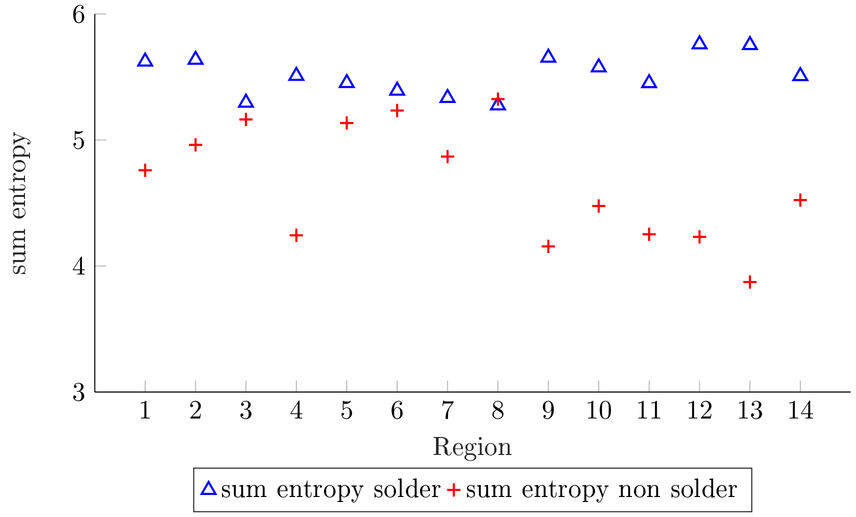


Figure A.7 Comparison of sum entropy values of solder paste (blue) and non-solder structures (red). Regions were randomly selected. The sum entropy of solder joints and non solder areas shows a significant overlap and thus cannot be used for classification.

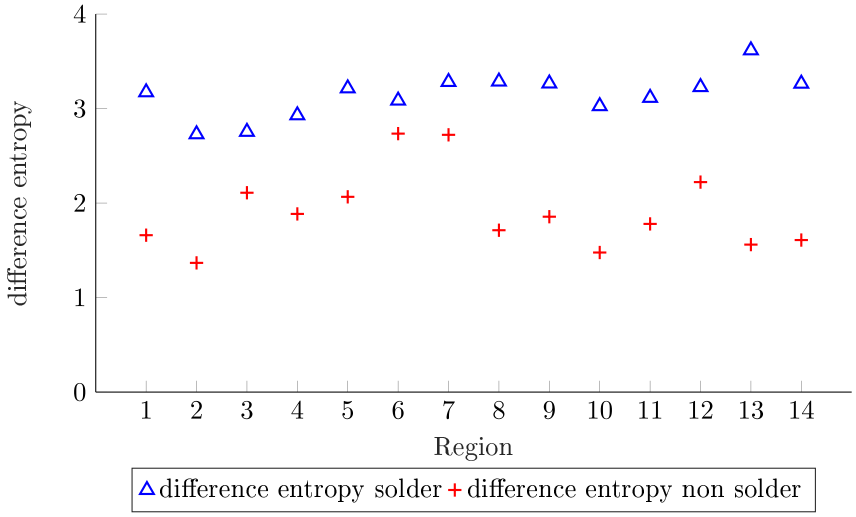


Figure A.8 Comparison of difference entropy values of solder paste (blue) and non-solder structures (red). Regions were randomly selected. The difference entropy show a slight overlap of solder joints and non-solder structures. Thus, it is not a proper feature to separate the structures.

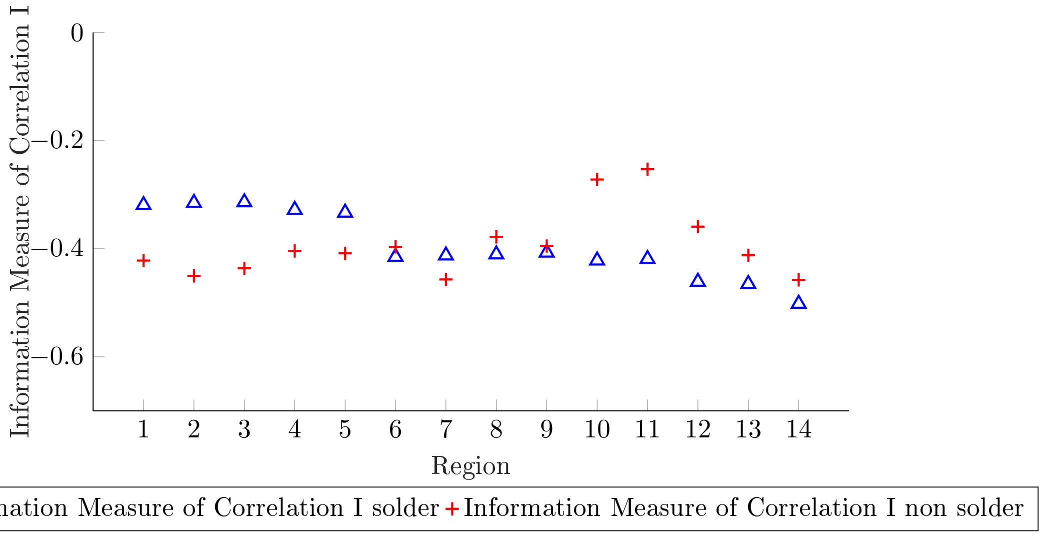


Figure A.9 Comparison of information measure of correlation 1 values of solder paste (blue) and non-solder structures (red). Regions were randomly selected. Due to a clear overlap in the data sets, this value is inappropriate for solder paste classification.

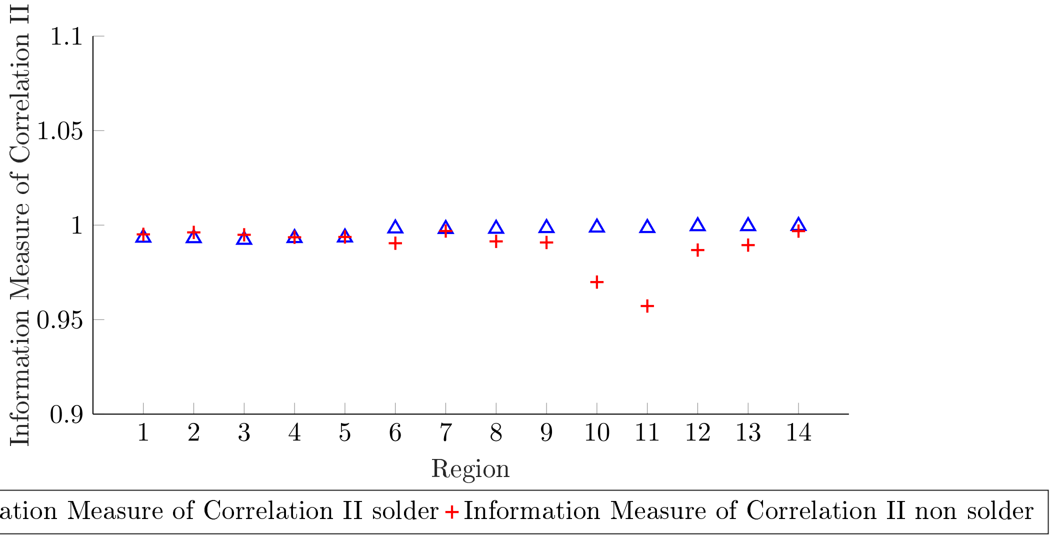


Figure A.10 Comparison of information measure of correlation 2 values of solder paste (blue) and non-solder structures (red). Regions were randomly selected. Due to a clear overlap in the data sets, this value is inappropriate for solder paste classification.

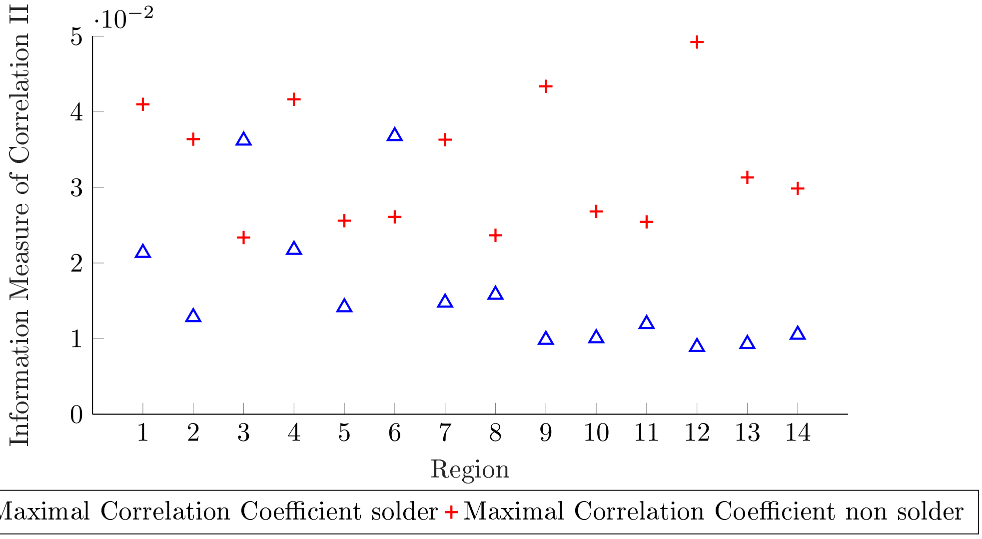
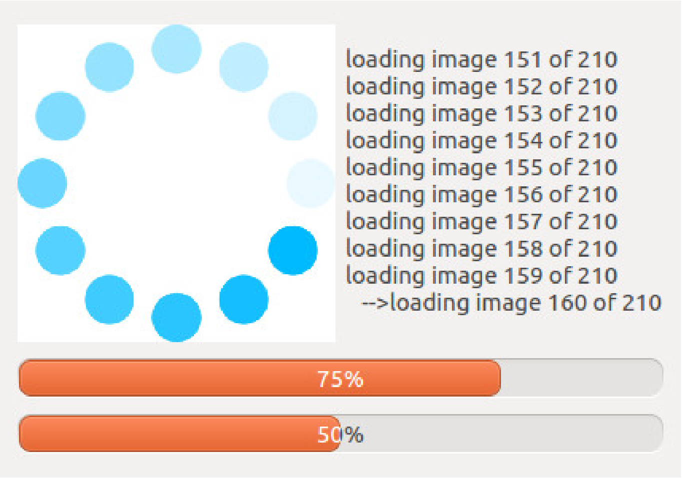


Figure A.11 Comparison of maximal correlation coefficient of solder paste (blue) and non-solder structures (red). Regions were randomly selected. The maximal correlation coefficient of solder joints and non-solder structures has a strong variance resulting in an overlap of the data sets. The value is improper for classification.

A.3 SolderInspect - Frontend

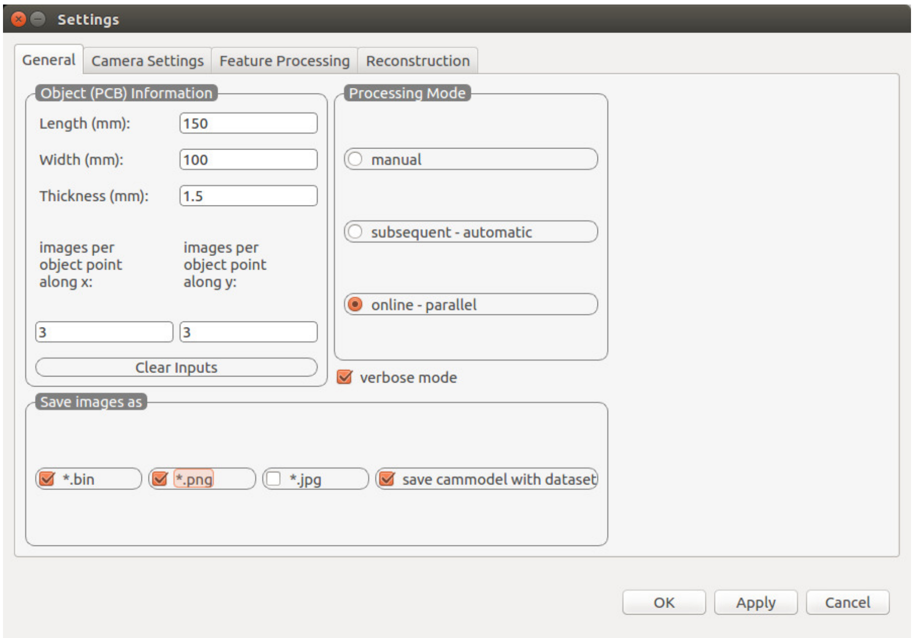


loading image 151 of 210
loading image 152 of 210
loading image 153 of 210
loading image 154 of 210
loading image 155 of 210
loading image 156 of 210
loading image 157 of 210
loading image 158 of 210
loading image 159 of 210
-->loading image 160 of 210

75%

50%

Figure A.12 SolderInspect GUI: Status of loading images when verbose mode is selected.



Object (PCB) Information

Length (mm): 150

Width (mm): 100

Thickness (mm): 1.5

images per object point along x: 3

images per object point along y: 3

Processing Mode

○ manual

○ subsequent - automatic

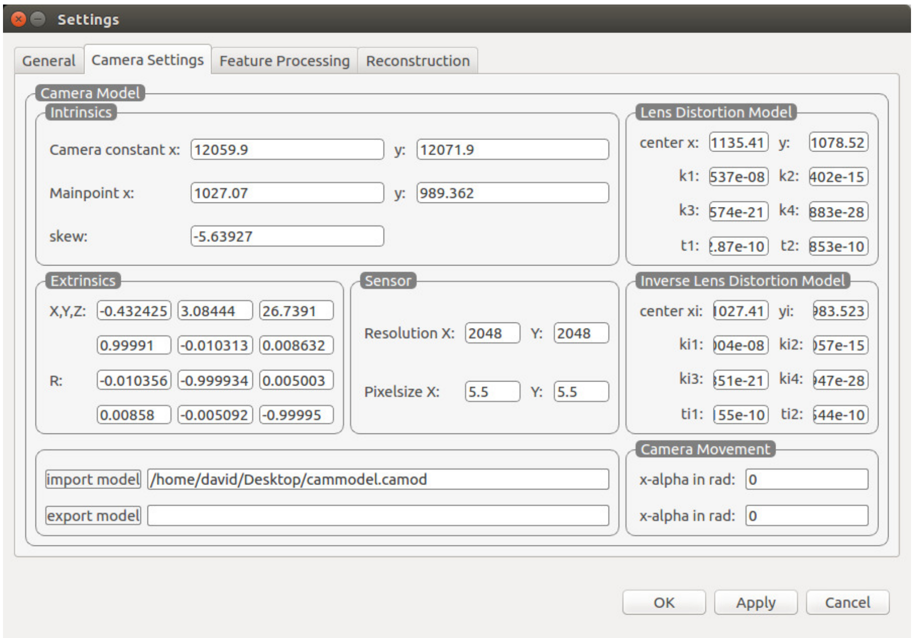
● online - parallel

☑ verbose mode

Save images as

☑ *.bin ☑ *.png ☐ *.jpg ☑ save cammodel with dataset

Figure A.13 SolderInspect GUI: General Settings - For a proper processing the width height and length of a PCB is needed. The user can define how many images per object position in x and y direction should be taken as well as the processing chain (manual, automatic, online).



Camera constant x: 12059.9 y: 12071.9
Mainpoint x: 1027.07 y: 989.362
skew: -5.63927

center x: 1135.41 y: 1078.52
k1: 537e-08 k2: 402e-15
k3: 574e-21 k4: 883e-28
t1: .87e-10 t2: 853e-10

X,Y,Z: -0.432425 3.08444 26.7391
R:
0.99991 -0.010313 0.008632
-0.010356 -0.999934 0.005003
0.00858 -0.005092 -0.99995

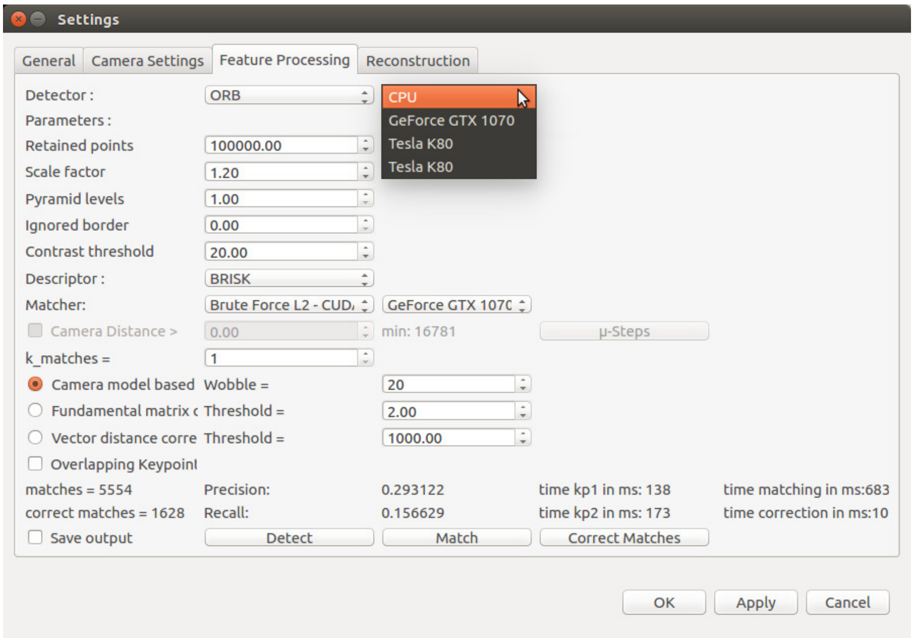
Resolution X: 2048 Y: 2048
Pixelsize X: 5.5 Y: 5.5

center xi: 1027.41 yi: 83.523
ki1: 04e-08 ki2: 57e-15
ki3: 51e-21 ki4: 47e-28
ti1: 55e-10 ti2: 44e-10

import model /home/david/Desktop/cammodel.camod

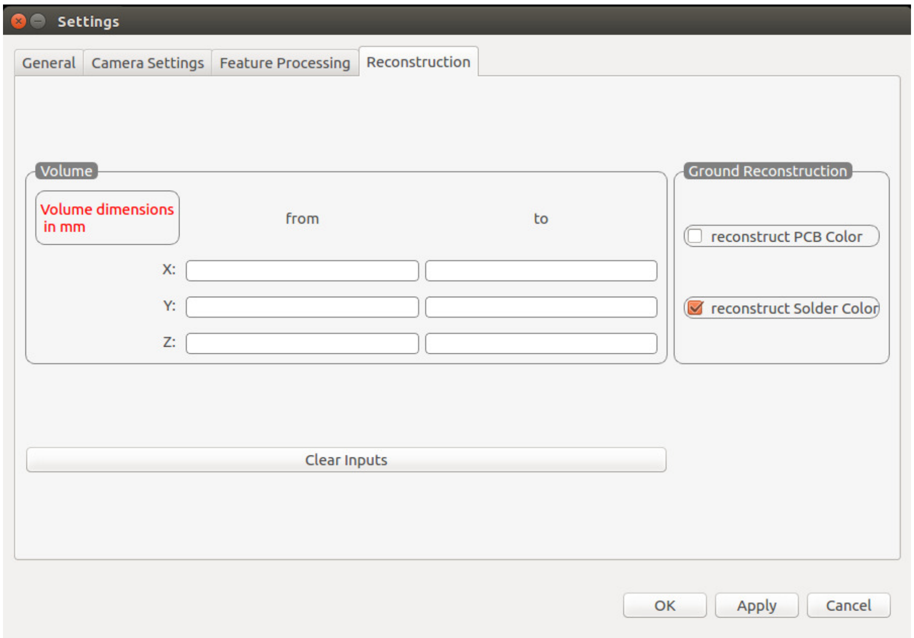
x-alpha in rad: 0
x-alpha in rad: 0

Figure A.14 SolderInspect GUI: Camera Model Settings - modification of camera model and im/export.



Detector : ORB
Parameters :
Retained points 100000.00
Scale factor 1.20
Pyramid levels 1.00
Ignored border 0.00
Contrast threshold 20.00
Descriptor : BRISK
Matcher: Brute Force L2 - CUD/ GeForce GTX 107(
☐ Camera Distance > 0.00 min: 16781
k_matches = 1
● Camera model based Wobble = 20
○ Fundamental matrix (Threshold = 2.00
○ Vector distance corre Threshold = 1000.00
☐ Overlapping Keypoint
matches = 5554 Precision: 0.293122 time kp1 in ms: 138 time matching in ms:683
correct matches = 1628 Recall: 0.156629 time kp2 in ms: 173 time correction in ms:10
☐ Save output

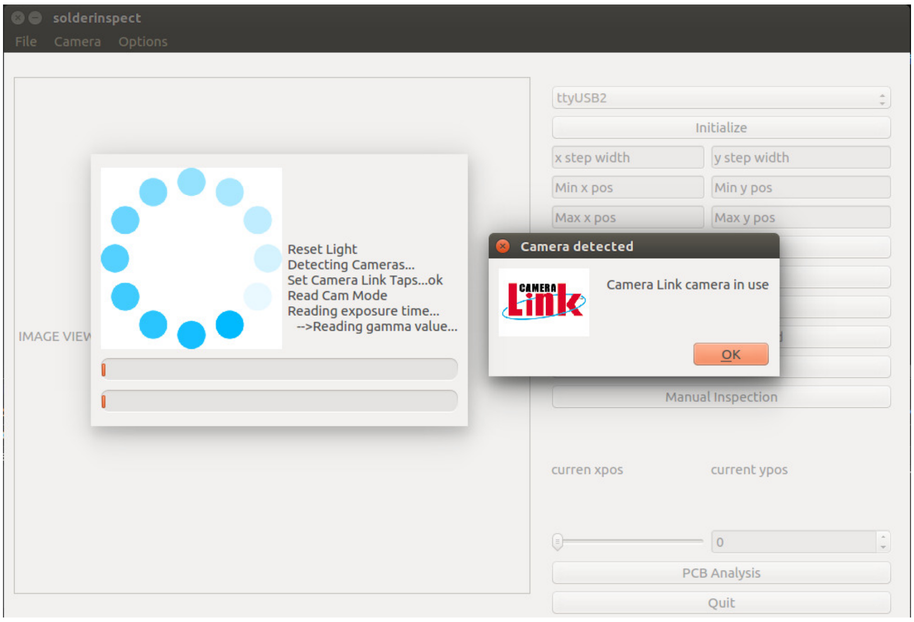
Figure A.15 SolderInspect GUI: Feature Settings - Settings for feature detectors, descriptors and matcher methods as well as hardware acceleration and matching correction.



☐ reconstruct PCB Color

☑ reconstruct Solder Color

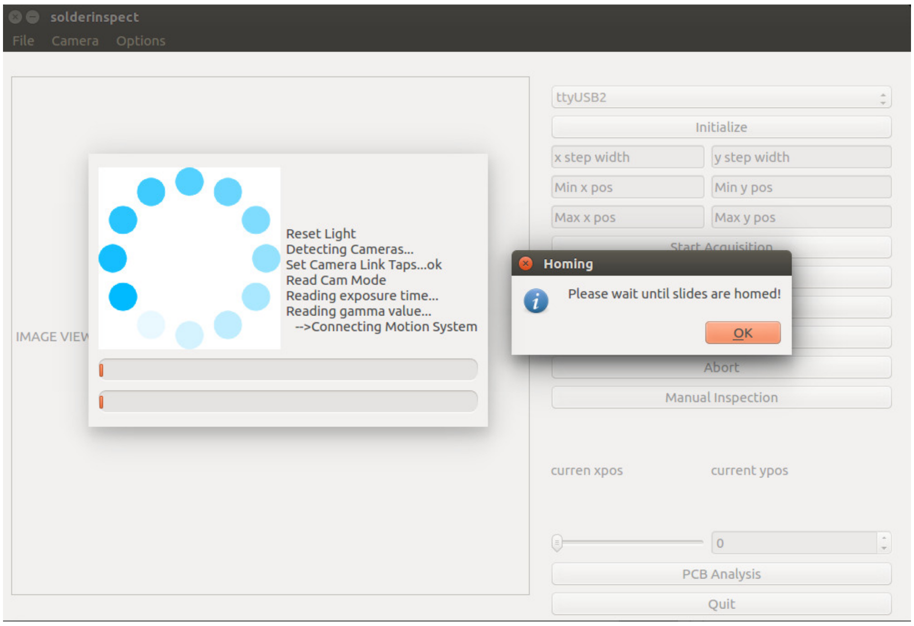
Figure A.16 SolderInspect GUI: Reconstruction Settings - These settings are optional. Definition of reconstruction volume, enabling Backprojection of whole images or only segmented solder joints.



```
Reset Light
Detecting Cameras...
Set Camera Link Taps...ok
Read Cam Mode
Reading exposure time...
  -->Reading gamma value...
```

Camera Link camera in use

Figure A.17 SolderInspect GUI: Initialization - camera detection and setup. The user is notified by this window which camera is detected and which parameters were read and set.

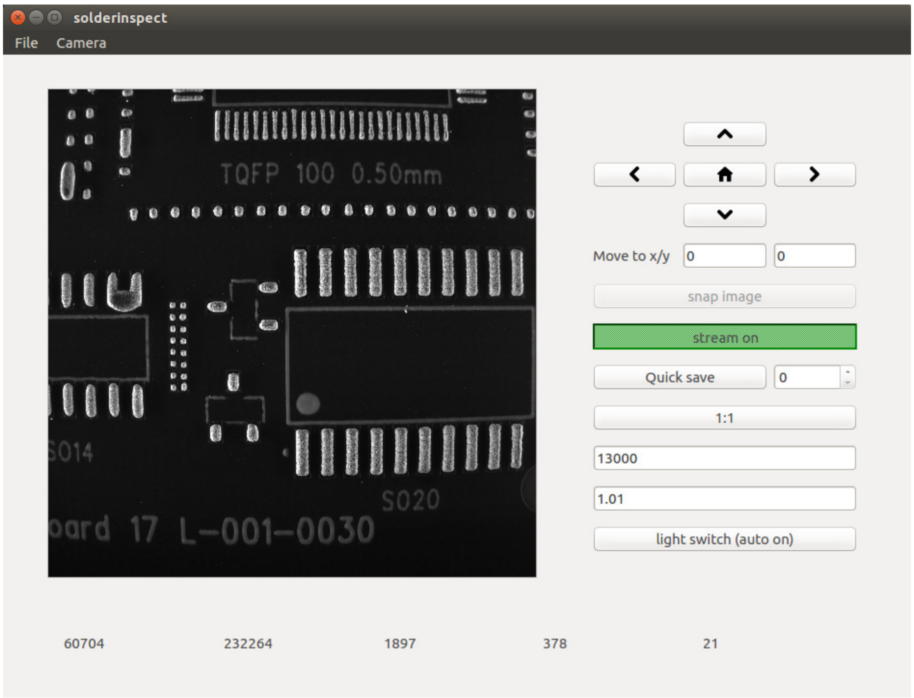


Reset Light
Detecting Cameras...
Set Camera Link Taps...ok
Read Cam Mode
Reading exposure time...
Reading gamma value...
-->Connecting Motion System

Homing

Please wait until slides are homed!

Figure A.18 SolderInspect GUI: Initialization - Homing: The user is notified that the linear slides are in homing preocess and is requested to wait until the home position is reached. Afterwards the main gui is enabled.



Move to x/y 0 0

Quick save 0

13000

1.01

60704 232264 1897 378 21

Figure A.19 SolderInspect GUI: Manual Inspection: For the examination of camera parameters and for manual inspection a manual inspection windows was created. The user can change camera settings(shutter speed, gamma, illumination brightness) and get a live view from the camera or a single image acquisition.

A.4 Feature Detection and Matching

A.4.1 SolFeC Example

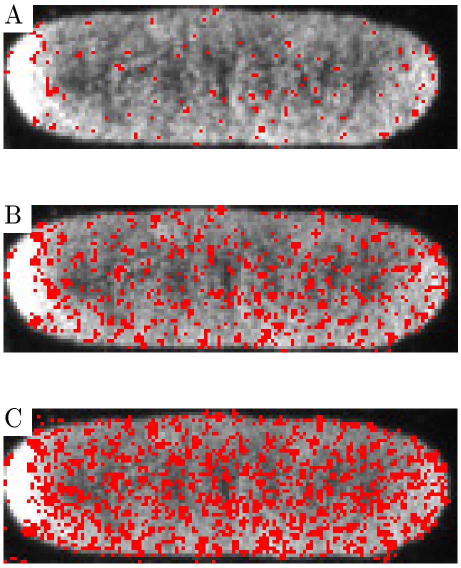


Figure A.20 Coverage of solder joints; A: small coverage with a SolFeC parameter of 0.16, B: medium coverage with a SolFeC parameter of 0.54, C: good coverage with a SolFeC parameter of 0.95.

A.4.2 Feature Detectors and Parametrization

MSER The computation time (see figure A.21) shows an exponential like decrease with an increase of the Δ gray level and a ranges from 1000 ms to 1950 ms. Even at lower coverage values a high time consumption is present. The number of keypoints show a correlation to coverage parameter. However, for higher Δ gray level, the ratio of solder joint keypoints to non-solder joint keypoints increases. This is due to the lower decrease of solder joint keypoints starting at Δ gray level 3 compared to the decrease of background keypoints as shown in figure A.22.

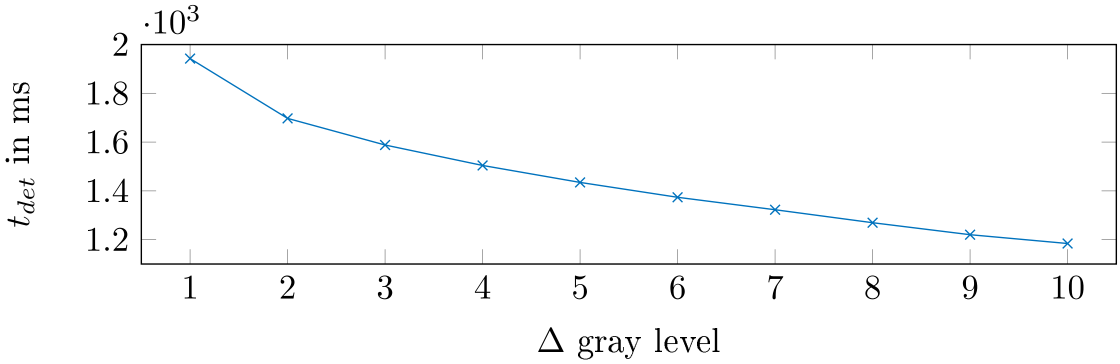


Figure A.21 Computation time consumption of MSER in relation to Δ gray level (Δ gray level= 1 - 10).

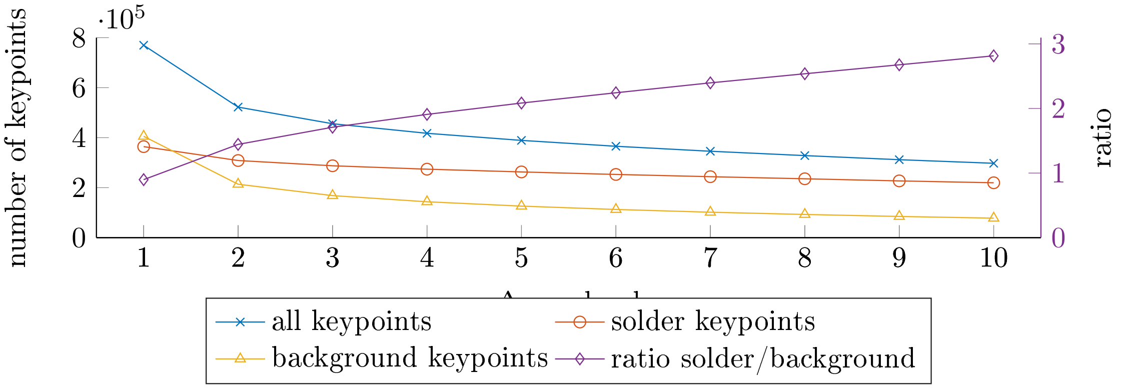


Figure A.22 Detected MSER Keypoints in relation to Δ gray level (Δ gray level= 1 - 10).

SIFT The computation time is mainly related to the number of octaves. More octaves need more time. However, this is different for the last two contrast threshold levels, where the computation time of 5 octaves drops below the one for 4 octaves (figure A.23). The other parameters show no influence on the detection time. A detection time of over 850 ms is needed to reach the best possible SolFeC parameter of 0.25.

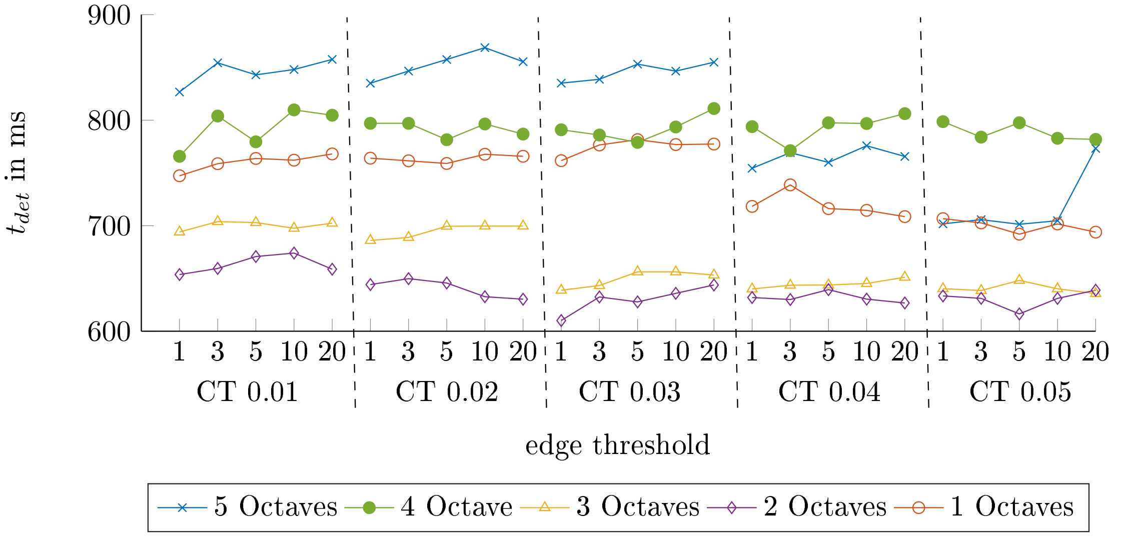


Figure A.23 Time consumption of SIFT for 5 different octaves, 5 contrast threshold (CT= 0.05, 0.04, 0.03, 0.02, 0.01) and 5 edge thresholds (20, 10, 5, 3, 1).

SURF Lower Hessian thresholds lead to a higher time consumption while less octaves and less layers per octave lead to a decrease of the detection time. For the best configuration (SolFeC = 0.11) a computation time of 780 ms is needed.

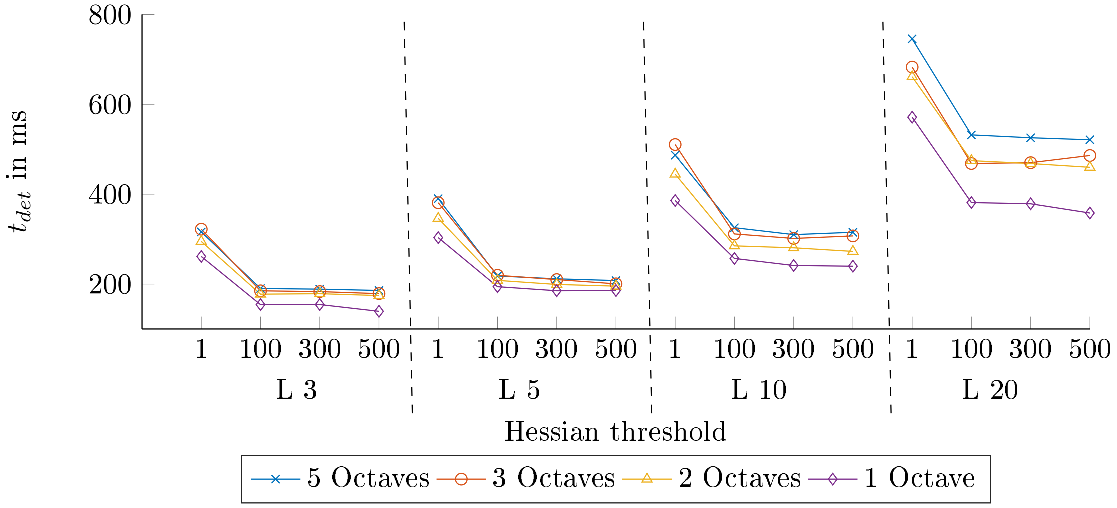


Figure A.24 Time consumption of SURF for 4 different octaves, 4 amounts of layers per octave (L = 20, 10, 5, 3) and 4 different hessian thresholds (500, 300, 100,1).